Semantically Enriched Text-Based Retrieval in Chemical Digital Libraries

Von der Carl-Friedrich-Gauß-Fakultät
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation

von Benjamin Köhncke
geboren am 15. Juli 1980
in Hannover

Eingereicht am:
17.09.2013

Disputation am:
18.12.2013

1. Referent:
Prof. Dr. Wolf-Tilo Balke

2. Referent:
Prof. Dr. techn. Wolfgang Nejdl

2014

Abstract

During the last decades, the information gathering process has considerably changed in science, research and development, and the private life. Whereas Web pages for private information seeking are usually accessed using well-known text-based search engines, complex documents for scientific research are often stored in digital libraries and will usually be accessed through domain specific Web portals. Considering the specific domain of chemistry, portals usually rely on graphical user-interfaces allowing for pictorial structure queries. The difficulty with purely text-based searches is that information seeking in chemical documents is generally focused on chemical entities, for which current standard search relies on complex and hard to extract structures.

In this thesis, we introduce a retrieval workflow for chemical digital libraries enabling text-based searches. First, we explain how to automatically index chemical documents with high completeness by creating enriched index pages containing different entity representations and synonyms. Next, we analyze different similarity measures for chemical entities. There are many different measures available, all of them relying on fingerprint representations of chemical entities. Our evaluations clearly show that many combinations are uncorrelated and that it is not possible to assign these uncorrelated combinations to specific chemical search tasks. The reason is that each chemist has specific background knowledge influencing his/her subjective perception of relevance. In order to model such implicit knowledge, we cluster chemical entities based on their functional groups. Using the functional groups clusters for retrieval, we are able to reduce the size of the result set by up to 90% without losing quality.

Furthermore, since users often search for chemical entities occurring in a specific context, we also show how to use contextual information to further enhance the retrieval quality. We present two different approaches: The first uses a similarity measure composed of different features gathered from the Wikipedia pages of the chemical entities. The resulting similarity measure combines context and entity similarity leading to improved retrieval quality compared to standard text-based search. In the second approach, we annotate chemical documents with cross-domain context terms. We use documents from the related domain of biomedicine, which are annotated with terms from the MeSH ontology. Then, we learn classification models based on the contained chemical entities and automatically annotate chemical documents with these terms. To assure that the associated terms are semantically related to the content of the documents, we use Wikipedia as a semantic filter and remove all unrelated terms. Our experiments prove the usefulness of cross-domain ontology terms for improving the retrieval quality for contextual search in chemistry.

However, the annotated terms will not help for contextual search if the users use different vocabulary than provided by the annotated terms. Therefore, we present an approach that semantically enriches documents with Wikipedia concepts to overcome the vocabulary problem. Our evaluations show that the provided approach

outperforms state-of-the-art query expansion and Latent Semantic Analysis methods. With Wikipedia we are able to bridge the gap between the context terms provided by the users and the vocabulary used in the documents. Since for most queries a huge amount of possibly relevant hits are returned to the user, we further present an approach summarizing the documents' content. Each document is represented as a tag cloud consisting of its associated Wikipedia categories. Our experiments with domain experts show that Wikipedia categories are even more useful to describe chemical documents than terms from a domain-specific ontology. Thus, we can state that the Wikipedia categories system can be used in domain-specific portals to overcome the problem of expensive, manually created ontology knowledge.

Finally, we present an architecture for a chemical digital library provider combining the different steps enabling semantically enriched text-based retrieval for the chemical domain.

Zusammenfassung

Über die letzten Jahre hat sich der Prozess der Informationssuche stark verändert. Während im privaten Bereich meistens über eine text-basierte Websuche auf Informationen zugegriffen wird, erfolgt der Zugriff auf Dokumente für den wissenschaftlichen Gebrauch in der Regel über domänenspezifische Web Portale. Betrachtet man beispielsweise die Domäne der Chemie, basieren Web Portale auf speziellen grafischen Benutzeroberflächen, die gezeichnete, strukturbasierte Anfragen ermöglichen. Da die Informationssuche für chemische Dokumente generell auf chemischen Entitäten basiert, die wiederum aus komplexen Strukturen bestehen, birgt eine reine text-basierte Suche eine Vielzahl von Herausforderungen.

In dieser Arbeit stellen wir uns diesen Herausforderungen und entwickeln einen Retrieval Workflow für eine chemische digitale Bibliothek, der text-basierte Suchen ermöglicht. Als erstes erläutern wir wie man chemische Dokumente automatisch indexiert, indem wir Indexseiten erzeugen, die mit semantischen Informationen angereichert werden. Diese Seiten beinhalten für jede chemische Entität des Dokumentes alle Synonyme, sowie unterschiedliche Repräsentationsmöglichkeiten. Im Folgenden erklären wir wie man Ähnlichkeit zwischen chemischen Entitäten bestimmen kann. In der Chemie gibt es eine Vielzahl von Ähnlichkeitsmaßen. Alle haben gemein, dass sie auf einer Fingerprint Darstellung der chemischen Entitäten basieren. Unsere Auswertungen mit den verschiedenen Maßen zeigen, dass viele unkorreliert sind. Darüber hinaus wird klar, dass diese unkorrelierten Maße nicht zu spezifischen Suchtasks der Chemie zugeordnet werden können. Der Grund ist, dass Chemiker spezielles Hintergrundwissen haben, das ihr subjektives Relevanzempfinden beeinflusst. Um dieses Relevanzempfinden zu modellieren fügen wir chemische Entitäten basierend auf ihren funktionellen Gruppen zu Clustern zusammen. In dem wir diese Cluster für das Retrieval verwenden, sind wir in der Lage, die Größe der Ergebnismenge um bis zu 90% zu reduzieren, ohne jedoch die Retrievalqualität zu verschlechtern.

Im Anschluss beschäftigen wir uns mit der Tatsache, dass Benutzer häufig nach chemischen Entitäten suchen, die in einem bestimmten Kontext auftreten. Es ist wichtig diese Kontextinformation zu berücksichtigen um die Retrievalqualität weiter zu verbessern. Wir präsentieren zwei verschiedene Ansätze um kontextuelle Suche zu ermöglichen. Im ersten Ansatz entwickeln wir ein Ähnlichkeitsmaß, das sich aus verschiedenen Features zusammensetzt, die wir aus den Wikipedia Seiten der jeweiligen chemischen Entitäten extrahieren. Das entstandene Maß kombiniert Entitäten- und Kontextähnlichkeit und führt zu einer verbesserten Retrievalqualität im Vergleich zur Standard Textsuche. Im zweiten Verfahren annotieren wir chemische Dokumente mit Kontext Termen, die wir aus verwandten Domänen beziehen. Wir nutzen Dokumente aus dem Bereich der Biomedizin, die wiederrum alle mit Termen aus der MeSH Ontologie annotiert sind. Im nächsten Schritt lernen wir Klassifikationsmodelle, basierend auf den in den Dokumenten beinhalteten chemischen Entitä-

ten. Mittels dieser Modelle werden chemische Dokumente automatisch mit den Termen der MeSH Ontologie annotiert. Um sicherzustellen, dass die annotierten Terme auch semantisch mit dem Inhalt der Dokumente zusammenhängen, nutzen wir Wikipedia als eine Art semantischen Filter und entfernen alle irrelevanten Terme. Unsere Experimente unterstreichen die Nützlichkeit dieser Annotationen für kontextuelle Suche in der Chemie.

Allerdings sind diese Terme nutzlos, falls die Benutzer ein völlig anderes Vokabular für ihre Kontextterme verwenden. Deshalb präsentieren wir einen Ansatz, der die Dokumente semantisch mit Wikipedia Konzepten anreichert um das Problem des unterschiedlichen Vokabulars zu beheben. Unsere Experimente zeigen, dass der vorgestellte Ansatz bessere Ergebnisse erzielt als Methoden basierend auf Query Expansion und Latent Semantic Analysis. Mit Wikipedia sind wir in der Lage die Lücke zwischen den gewählten Kontexttermen der Benutzer und dem Vokabular der Dokumente zu schließen.

Ein weiteres Problem, dem wir uns im Rahmen dieser Arbeit gestellt haben, ist, dass für die meisten Anfragen eine Vielzahl von möglicherweise relevanten Treffern zurückgeliefert wird. Deshalb präsentieren wir eine Methode um den Inhalt der Dokumente auf übersichtliche Weise darzustellen. Jedes Dokument wird als eine Tag Cloud bestehend aus Wikipedia Kategorien dargestellt. Interessanterweise zeigen unsere Versuche mit Domänenexperten, dass die Wikipedia Kategorien besser geeignet sind um chemische Dokumente zusammenzufassen als die Terme einer domänenspezifischen Ontologie. Daraus können wir folgern, dass die Wikipedia Kategorien eine gute Alternative für domänenspezifische Portale darstellen, um das Problem der teuren, manuell erzeugten Ontologien zu beheben.

Schlussendlich präsentieren wir eine Architektur für eine chemische digitale Bibliothek, die die gewonnenen Erkenntnisse kombiniert und semantisch angereicherte, text-basierte Suche in der Chemie ermöglicht.

Foreword

In this thesis, I present the work we did during the last 5 years in the area of chemical digital libraries. The ideas and approaches presented in this thesis have been published in various peer-reviewed conferences.

The work presented in Chapter 2 is based on the ideas published in:

- S. Tönnies, B. Köhncke, O. Koepler, and W.-T. Balke, "Exposing the Hidden Web for Chemical Digital Libraries," In *Proceedings of the10th ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, Surfers Paradise, Gold Coast, Australia, 2010.
- S. Tönnies, B. Köhncke, O. Koepler, and W.-T. Balke, "Building Chemical Information Systems - the ViFaChem II Project," In *Proceedings of Datenbanksysteme in Business, Technologie und Web (BTW),* 13. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS), GI, 2009.

Chapter 3 is built upon the work published in:

- B. Köhncke, S. Tönnies, and W.-T. Balke, "Catching the Drift – Indexing Implicit Knowledge in Chemical Digital Libraries", In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)*, Paphos, Cyprus, 2012.
- S. Tönnies, B. Köhncke, and W.-T. Balke, "Taking Chemistry to the Task – Personalized Queries for Chemical Digital Libraries," In *Proceedings of the 11th ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, Ottawa, Canada, 2011.

The contributions presented in Chapter 4 are published in:

- B. Köhncke, P. Siehndel, and W.-T. Balke, "Bridging the Gap - Using External Knowledge Bases for Context-Aware Document Retrieval", In *Proceedings of the 15th International Conference on Asia-Pacific Digital Libraries (ICADL)*, Bangalore, India, 2013.
- B. Köhncke, and W.-T. Balke, "Context-Sensitive Ranking Using Cross-Domain Knowledge for Chemical Digital Libraries", In *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)*, Valletta, Malta, 2013.

In Chapter 5 we present our research published in:

- B. Köhncke, and W.-T. Balke, "Using Wikipedia Categories for Compact Representations of Chemical Documents", In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, Toronto, Canada, 2010.

We also published other papers related to the area of digital libraries that are not used in this thesis:

- B. Köhncke, S. Tönnies, and W. - T. Balke, "Bi2SoN – A Digital Library for Supporting Biomedical Research", In *Proceedings of the 12th ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, Washington, DC, USA, 2012.
- S. Tönnies, B. Köhncke, W.-T. Balke, "Meta-Line: Lineage Information for Improved Metadata Quality", In *Proceedings of the 12th ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, Washington DC, USA, 2012.
- O. Koepler, W.-T. Balke, B. Köhncke, and S. Tönnies, "Personalized Information Spaces for Chemical Digital Libraries", Chemistry Central Journal, Vol. 3, 2009.
- O. Koepler, W.-T. Balke, B. Köhncke, and S. Tönnies, "Personalized Information Spaces: Improved Access to Chemical Digital Libraries," In *Proceedings of the 5th German Conference on Chemoinformatics*, Goslar, Germany, 2009.

Furthermore, the following publications have been published in the area of personalization and multimedia streaming in service-oriented architectures:

- S. Tönnies, B. Köhncke, P. Hennig, I. Brunkhorst, and W.-T. Balke, "A Service Oriented Architecture for Personalized Universal Media Access," Future Internet, Vol. 3, Apr. 2011, pp. 87-116.
- B. Köhncke, and W.-T. Balke, "MPEG-7/21: Structured Metadata for Handling and Personalizing Multimedia Content", The Handbook of MPEG Applications: Standards in Practice: Wiley, 2010.
- S. Tönnies, B. Köhncke, P. Hennig, and W.-T. Balke, "A Service Oriented Architecture for Personalized Rich Media Delivery," In *Proceedings of the IEEE International Conference on Services Computing (SCC)*, Bangalore, India, 2009.
- B. Köhncke, and W.-T. Balke, "Preference-Driven Personalization for Flexible Digital Item Adaptation", Multimedia Systems Journal (MMSJ), Vol. 13(2): Springer, 2007.
- B. Köhncke, and W.-T. Balke, "Personalized Digital Item Adaptation in Service-Oriented Environments", In *Proceedings of the 1st International Workshop on Semantic Media Adaptation and Personalization (SMAP)*, Athens, Greece, 2006.

Acknowledgements

First of all I would like to thank my supervisor Prof. Dr. Wolf-Tilo Balke for guiding me through the years of my thesis. He already introduced me to research when I was a student and finally also gave me the chance to work on this thesis. Thus, we have been working together for quite some years now and had a lot of interesting discussions. Without his support and guidance this thesis would not have been possible.

I would also like to thank Prof. Dr. Wolfgang Nejdl for being my second examiner. I loved the time working at the L3S Research Center on many interesting topics and together with many interesting colleagues.

I had a lot of interesting discussions and collaborations with my colleagues from the L3S Research Center and the Institute of Information Systems at the Technical University of Braunschweig. It was also very interesting to work in an international team and get to know many insights and meanings from different cultures. Especially, I would like to thank Sascha Tönnies for his ideas, collaboration, and for sharing the office with me for the last 5 years. Also a big thank to Oliver Koepler for many tips and insights about the domain of chemistry. Furthermore, there are many other colleagues I would like to thank for having a lots of fun together, also at shared activities outside the workplace, especially, Kerstin Bischoff, Ralf Krestel, Mohammad Alrifai, Patrick Siehndel, Ricardo Kawase, Silviu Homoceanu, Joachim Selke, and Christoph Lofi.

I would also like to thank my parents for their support and patience. Finally, I would like to thank my wife Panagiota for always supporting and loving me, and giving me power throughout the years of my thesis.

Table of Contents

Chapter 1

Introduction

In recent years, access to information and information gathering processes have changed in both academic as well as industrial research. Whereas libraries as physical institutions are losing importance, more and more information is made available online by specialized content providers like topical digital libraries, (open access) journals, and publishing houses. The connection between these providers is that digital libraries contain several (open access) journals and work together with the related publishing houses, which hold the copyright licenses. In general, based on the DELOS Digital Library Reference Model [1], a digital library is defined as:

"An organization, which might be virtual, that comprehensively collects, manages and preserves for the long term rich digital content, and offers to its user communities specialized functionality on that content, of measurable quality and according to codified policies."

In other words, digital library providers have to meet the following fundamental requirements to enable suitable retrieval. The documents stored in a digital library are typically part of the deep Web, meaning they cannot easily be found by search engine crawlers. Thus, to collect and manage the digital content, it is necessary to build proper indexes by extracting terms and enriching them with additional metadata. Moreover, the indexed data has to be complete and of high quality. To offer appropriate access to the indexed data, it is important to define ranking functions and to offer comprehensible representations of search results.

It is interesting to see that besides the common catalog-based searches for literature and mostly bibliographic information, digital library providers nowadays also extend their services to add value. In particular, personalizable portals enabling user-centered searches over heterogeneous document collections and databases are currently gaining momentum. Indeed, consumers may have different workflows and expectations when searching for relevant literature, strongly depending on the scientific domain, the level of expertise, and the task at hand. There are many different topic-centered providers, offering access to domain specific literature. In the course of this thesis, we choose chemistry as an example domain.

1.1. Foundations of Chemical Search

In the domain of chemistry, information seeking is essentially centered on chemical entities. The usual representation of chemical entities is often based on chemical structures, which are somehow embedded (often as drawn structures) into the documents. Thus, to assure the high quality requirements of a chemical digital library, it is necessary to extract and index chemical entities from documents. The problem is that graphical representations of chemical entities still cannot be easily transferred

into the digital world once published in a document. Whereas domain experts can easily identify the shown structures and classify them in the context of the document, it is currently impossible to extract this information automatically while meeting the high quality standards of a digital library. Over the last years, several projects, like CLiDE Pro[1] or chemoCR[2], focused on developing a chemical optical recognition for the reconstruction of chemical structure information from digitized documents. However, recognition rates always have proven to be insufficient for a production environment [2], [3], [4], [5].

Facing these problems, service provider for chemical information offer specialized indexes. These indexes are built by manually identifying and indexing all chemical structures from a document collection in structure databases. The amount of manual work required for building and maintaining such indexes results in high costs. Today, the most important provider in the domain of chemistry is the Chemical Abstract Service (CAS). CAS as a subsidiary of the American Chemical Society (ACS) offers a specialized digital library indexing a variety of chemical document collections. The CAS Registry, as addition to the CAS database, was already introduced in 1965 to overcome problems with identifying chemical entities based on their names. Since digital libraries promise high quality information access, the ACS is maintaining their entity database by manually indexing all chemical entities occurring in journal articles, conferences, patents, and many other research publications in the chemical domain. Further, they annotate the documents in order to build their CAS search index for chemical literature, resulting in a high quality digital library. This quality is gained at the expense of high costs for the manual indexing process. For each chemical entity, approximately three Euros have to be spent to fully store relevant information in the CAS registry, when extracted from literature and correctly drawn by a domain expert for a structure database. Currently the CAS registry comprises over 73 million substances, but search engine access is very expensive and strictly restricted to subscribers at a price starting from 2,600 USD/year for a single user subscription dependent on the size of the company.

There are also first approaches for automatically extracting *textual* representations of chemical entities from documents to overcome the expensive manual extraction process. Prime examples are the OSCAR framework [6], [7] and ChemSpot [8]. OSCAR can identify and extract multiple name variations of chemical entities. In combination with name-to-structure algorithms these entity names can also be transformed into chemical structure information. Of course, the quality of automatic entity extraction is not as good as manual annotations. Dependent on the query term the recognition rates of OSCAR vary between 69% to 81% for recall and 64% to 75% for precision [6]. The ChemSpot framework was evaluated on the SCAI corpus [9] and reaches an average recall of 71.9% and an average precision of 76.6%.

[1] www.keymodule.co.uk/CLiDE.html

[2] www.scai.fraunhofer.de/chemocr.html

Nevertheless, compared to expensive manual annotations these frameworks are helpful alternatives to extract chemical terms.

Access to chemical information is usually performed through graphical interfaces. By drawing a chemical structure, a domain expert can thus formulate a query, which in turn will be parsed by the chemical query parser and matched against entities' fingerprints stored inside a structure database. Already during the 19th century, inspired by the work of Jacob H. van't Hoff and August Kekulé, drawings of chemical structures became the common way of communicating chemical information about substances and their reactions. Today, we speak of chemical structure representations as the 'language of chemists' [10]. The chemical structure is a simple to understand, yet most precise way to uniquely describe a chemical entity, leaving the ambiguity of systematic, IUPAC[3], trivial, or brand names behind. Graphical representations of chemical entities are therefore commonly used as query terms when searching for chemical information. However, chemists use different types of (graphical) queries to search for information.

The most common type is a chemical **substructure search**, where all molecules from a database are retrieved that contain a user-defined query substructure independent of the context this structure occurs in [11]. However, substructure searching has several limitations arising from the requirement that each retrieved molecule from the database has to contain the entire query substructure [12]. That means that while posing his/her query the user already needs a very clear view of the possible types of structures that will be retrieved [13]. Imagine a chemist from the area of drug design, posing a pharmacophore query. A pharmacophore is defined by the IUPAC as follows [14]:

"A pharmacophore is the ensemble of steric and electronic features that is necessary to ensure the optimal supramolecular interactions with a specific biological target structure and to trigger (or to block) its biological response. A pharmacophore does not represent a real molecule or a real association of functional groups, but a purely abstract concept that accounts for the common molecular interaction capacities of a group of compounds towards their target structure. The pharmacophore can be considered as the largest common denominator shared by a set of active molecules."

For such a query the user needs sufficient knowledge about the geometric requirements for the activity he/she is interested in, to retrieve only those molecules fitting in the biological receptor site. Usually such pharmacophores are found by comparing several bioactive molecules and identify the features they have in common [15]. Obviously, at the beginning of an investigation it is very difficult to specify suitable features that are responsible for the observed activity.

Furthermore, the user has no control about the size of the output of a substructure query. A broadly defined query, e.g., containing a common ring system, can

[3] http://www.iupac.org

retrieve thousands of retrieval results. Usually the user has the possibility to filter the result set afterwards or to refine his/her query to get a suitable amount of results for subsequent analysis. But finally, a substructure query always partitions the set of available chemical entities in the database into two distinct subsets, i.e., those chemical entities containing the query substructure and those that do not. Especially for queries aiming at finding possible bioactives in the database, there is no suitable way to rank the retrieved chemical entities using substructure search.

These limitations of substructure search have led to the development of **similarity search** [13]. Here, a query consists of an entire chemical entity instead of just a substructure. There are many different similarity measures available. All of them rely on fingerprint representations of chemical entities. A fingerprint is a sequence of bits, where each bit is set according to the occurrence of specific chemical features. There are several fingerprint representations available encoding different types of chemical features (see Chapter 3.1). Using similarity search, the similarity between a query and each chemical entity in the database can be computed leading to a ranked list of retrieval results.

However, although in similarity search the results are ranked, the amount of possibly relevant chemical entities is still high. One possibility to restrict the result set and to get more focused results is to use **contextual search**. When users search for chemical entities, they are often interested in similar entities occurring in a specific context. It is important to consider this context because the similarity of two chemical substances actually depends heavily on the search context. Consider, for instance, the chemical entities *Zanamivir* and *Ibuprofen*. Both are used in the treatment of flu and are therefore similar regarding this pharmacological activity context. *Ibuprofen* is also used to treat inflammatory diseases such as rheumatoid arthritis. However, regarding this context both entities are very dissimilar: *Zanamivir* is a neuraminidase inhibitor and thus not in the least useful for the treatment of rheumatoid arthritis. It is therefore necessary to personalize measures for entity similarity to the task or search context a user is currently engaged in. In brief, context used to disambiguate the user's explicit query can be expected to lead to focused and relevant retrieval results.

However, also in highly specialized domains like chemistry an exponential growth of available information can be observed. A good example for the information growth in this domain is shown in a press release of the CAS. Today, more than 73 million unique chemical substances have been indexed in the curated CAS registry, the worldwide most comprehensive registry of chemical substances. Remarkable is that only 10 million substances have been indexed in 1990, meaning that the amount of chemical entities doubles every ten years. In addition, since more and more publishers offer digital access to their data, also the amount of available publications in the Web is increasing fast. Recently, huge providers offering access to open access literature have started their services, like, for example, the Public Library of Science

(PLOS)[4] or the Multidisciplinary Digital Publishing Institute (MDPI)[5]. Since the funding for the open access initiative is guaranteed, this trend will further increase during the next years, see, e.g., the Horizon 2020 program of the EU [16]. Of course, also the number of chemical publications is growing fast. Currently there are 189 chemistry journals listed in the Directory of Open Access Journals (DOAJ[6]). It is important to open up the knowledge of these sources to practitioners in the chemical domain. Without suitable indexing and search interfaces, the contained documents are not easily detectable. Obviously, for this growing open access movement, indexing strategies and query interfaces as offered by CAS are quite too expensive and therefore not a viable option.

Thus, several groups are currently working on building free high quality chemical search engines to overcome the costly access to chemical literature. Prime examples are the substance database PubChem[7] combining several chemical entity data sources and the document search engine ChemXSeer [17]. ChemXSeer relies on a highly complex process extracting chemical formulae in an automated way and linking them to the documents. Other platforms, like ZINC[8], ChemBank[9], or ChemDB [18] provide detailed information about some chemical structures, names, and properties. Another promising search platform is the ChemSpider portal[10], which is maintained by the Royal Society of Chemistry (RSC). ChemSpider has started to connect the knowledge contained in different information sources. For example, chemical entities are connected to their Wikipedia page and the respective terms from the MeSH ontology.

However, of course, not only experts, who have access to domain specific search engines, are interested in chemical entities. Most users often use a **text-based search**, e.g., using Google, as starting point for their information gathering process. In August 2012, more than 11 billion queries have been posed in the US using Google[11]. Therefore, it is important to also make documents stored in domain-specific digital libraries detectable via text-based queries. Moreover, text-based queries have several advantages compared to structural queries. Chemists can also search for brand names, like, e.g., Viagra, instead of drawing the structure of some active ingredient, like in this case Sildenafil. Furthermore, in chemical structure search it is not possible to submit Boolean queries. To search for documents containing two or

[4] http://www.plos.org

[5] http://www.mdpi.com

[6] http://www.doaj.org

[7] http://pubchem.ncbi.nlm.nih.gov

[8] http://zinc.docking.org

[9] http://chembank.broadinstitute.org

[10] http://www.chemspider.com

[11] http://www.comscore.com

more chemical entities two subsequent structure queries are necessary. Thus, the retrieval time, which is already at least five times higher compared to text-based retrieval, is further increased (see evaluations in Chapter 2.3.3). Also from the viewpoint of information providers, suitable text-based retrieval is desirable to avoid complex graphical query interfaces and to offer faster retrieval. Most information providers in chemistry already offer basic functionalities for text-based retrieval. ChemXSeer offers access to chemical literature using a search index based on chemical formulae. While from the view of computer scientists some challenging problems had to be solved to extract chemical formulae from documents and built proper indexes [19], [20], they are not useful for chemists due to their ambiguity.

However, the basics to enable text-based search have already been defined: besides chemical structures, there are several other useful, textual representations of chemical entities. For a long time, chemists have developed complex algorithms for converting a chemical structure into unique line notations. Such a notation is, e.g., the IUPAC name which yields into a unique representation for small molecules (introduced around 1920). But, for more complex molecules, the IUPAC rules prove ambiguous. Particularly, for use in digital systems chemical names have been transformed into linear notations. Today, the prevalent linear notations are the International Chemical Identifier (InChI) and the simplified molecular input line entry specification (SMILES), which both indeed are unique representations, but show high complexity and are almost impossible to read for humans. Therefore, they are not widely used in chemical documents and thus cannot be readily extracted for indexing purposes. Thus, a straightforward keyword-based access like provided by common search engines such as Google or Yahoo!, is still insufficiently supported for Web pages dealing with chemical information. Therefore, the goal of this thesis is to enable semantically enriched text-based retrieval in chemical digital libraries.

1.2. Problem Statement and Thesis Structure

Today, a lot of chemical information is available online, e.g., in open access journals and topical databases. It is important to open up this knowledge and make it available to practitioners from the chemical domain. The goal is to store the available information in a chemical digital library, which offers suitable access for domain experts. To reach this goal, many different steps are necessary, starting with extracting chemical entities and proper indexing, over finding suitable similarity measures and rankings, to defining comprehensible representations of query results (see **Fig. 1**). Furthermore, when searching for literature, chemists use different types of queries that need to be supported by a chemical digital library provider, i.e., structure-based queries, text-based queries, and contextual queries. Whereas for structure-based queries already a lot of work has been done regarding indexing chemical entities and storing them in specific structure databases, the other query types are still insufficiently supported. This leads to a number of challenges for the chemical domain, which we tackle in this thesis:

- Building suitable indexes to enable text-based retrieval,
- Analyzing different similarity measures available for structural queries,
- Finding approaches enabling contextual search in the domain of chemistry,
- Developing comprehensible representations for search results,
- Integrating steps, like semantic metadata enrichment, in the processing workflow of a chemical digital library.

Since for structural queries approaches are already available, the main focus of this thesis is to provide more advanced approaches to enable semantically enriched text-based retrieval in chemistry. **Fig. 1** gives an overview of a chemical digital library workflow. We will explain each step in detail in the following chapters.

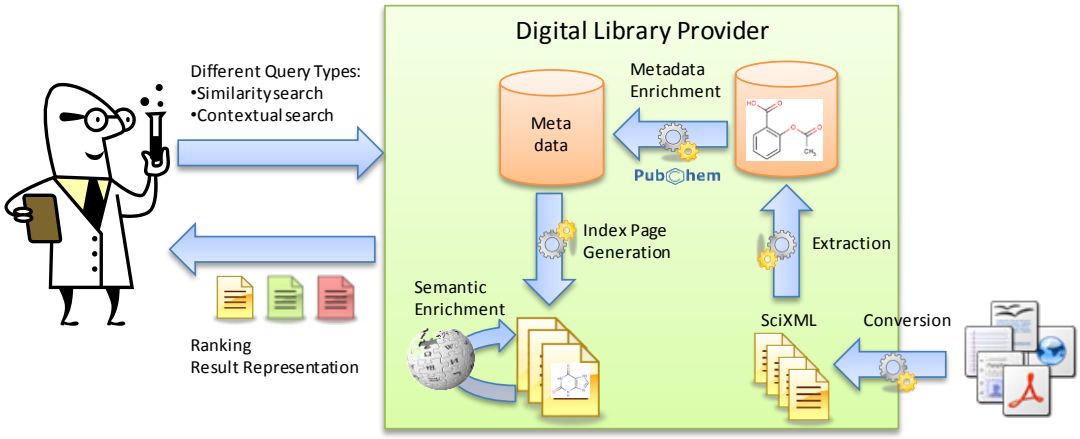


Fig. 1. Chemical digital library workflow

- The first step to enable textual queries lies in proper indexing of chemical documents. Therefore, we will show how to convert all documents into a uniform interface format. Afterwards, we extract the chemical entities from the documents and enrich them with synonyms and different entity representations. Finally, we create enriched index pages allowing for suitable text-based retrieval. The details are explained in Chapter 2.
- Since users are interested in finding similar entities regarding their query, we analyze and compare different similarity measures in Chapter 3. Our experiments show a lot of uncorrelated measures. It is not possible to assign a specific search task to a particular similarity measure. The reason is that each chemist has personal background knowledge influencing his/her perception of relevance. Therefore, we also show how to model such implicit knowledge of a chemist by clustering chemical entities based on their reaction behavior and the chemists' implicit understanding of chemical classes.
- Furthermore, chemists are not only searching for similar entities, but for entities occurring in certain contexts. In Chapter 4 we present two approaches enabling contextual search by using external knowledge bases. In the first approach, we

develop a similarity measure combining context- and entity similarity. In the second approach, we annotate chemical documents with cross-domain ontology terms leading to highly improved context-based retrieval. Since the annotated context terms are only useful if the users use the same vocabulary as provided by the terms, we show how to use Wikipedia to bridge the gap between the query context and the documents' vocabulary.

- Since usually a huge amount of results matching the user's query are returned, we show in Chapter 5 how to present search results to the user by using external knowledge to create tag cloud representations of documents.
- In Chapter 6 we present an architecture for a chemical digital library combining all these steps. We explain which data can already be pre-processed by a digital library provider. In addition, we explain in detail how a query is processed and how the retrieval quality can be further improved by using a system based on user feedback.
- Finally, in Chapter 7 we give a conclusion and an outlook to future work.

Chapter 2

Basic Indexing for Text-Based Retrieval

In this chapter, we explain how to make the large body of chemical knowledge stored in the Web widely searchable and accessible using text-based retrieval, however, with a minimal amount of manual indexing. Firstly, we show how to convert chemical documents into a general interface format and extract the contained chemical entities. Secondly, we present an information service that automatically generates enhanced metadata representations from chemical documents. These metadata enrichments include extensive information for each entity found in the documents' fulltexts, e.g., trivial names with synonyms, InChI codes, SMILES, and basic chemical properties. By generating respective HTML pages and linking to the respective document sources, current Web crawlers can easily index the information in connection with each document. To prove the quality of the generated enriched index pages, we compare them to chemical structure searches in a typical retrieval scenario. Our experiments clearly show the added value for chemical document retrieval. By providing rich and diverse metadata, our system is able to support typical, and even sophisticated chemical workflows. In contrast, previous approaches in digital libraries, like, e.g., indexing entities by simple chemical formulae, see [19], are entirely useless from a chemist's point of view due to the ambiguities: for instance for the simple formula C_6H_6 there are already more than 200 different structures, each of them with different chemical properties and uses.

To summarize, the different steps we explain in this chapter are:

- Conversion of chemical documents into a general interface format.
- Extraction of chemical entities.
- Enriching chemical entities with different representations and synonyms.
- Creation of enriched index pages.
- Proving the quality of the enriched index pages by comparing them to chemical structure searches.

2.1. Document Conversion and Entity Extraction

Digital library providers usually receive documents from many different repositories, showing a plethora of document styles, layouts and file formats. To enable suitable retrieval, the first necessary step is to convert the chemical documents into a general interface format. The standard choice is SciXML, which is a canonical XML format designed to represent the common hierarchical structure of scientific articles and is originally described in [21]. Its latest implementation, SciXML-CB, is based on an analysis of XML actually generated by scientific publishers in the fields of Chemistry

and Biology [22]. Whereas it is rather trivial to convert structured document formats, e.g., XML or HTML into the respective SciXML representation, the reality is different: most open access journals have only a PDF document collection. However, PDF documents are unstructured and do not lend themselves easily to content extraction. For instance, PDF documents store all characters using the absolute position within the document and thus all paragraphs are split during OCR processes into single line paragraphs. Since entity names usually are quite long, the probability that names are split into several parts by the OCR process is rather high. Thus, entity extractors have a hard time figuring out whether different parts belong to the same entity or are entities in their own right. Imagine the chemical name 4-(aminomethyl)cyclohexamine separated into 4-aminomethyl and cyclohexamine. In addition, subscript and superscript letters are important in chemical formulas and names, thus, extracting them correctly is essential. For instance, the chemical name (1,7,7)-Trimethyl-tricyclo[2.2.1.$0^{2,6}$]heptan is not a valid name without the superscript letters 2,6. As a last step, text fragments from tables and figures have to be removed.

After the conversion of all documents into SciXML, in the next step the chemical entities have to be extracted. In fact, the recognition of *named entities* is a major step in preprocessing and indexing not only for chemical documents. Natural language processing (NLP) techniques for named entity recognition are a highly active research area. For example, in the bioinformatics domain, a lot of publicly available resources are already in place, e.g., the well-known PubMed / Medline corpus or the manually annotated corpora generated by the PennBioIE[12] and GENIA[13] groups. In contrast, the development of NLP methodologies in the field of chemistry lags behind. In [19] an approach automatically extracting chemical formulae from documents is presented. The authors propose all necessary steps to build a chemical formula search engine. In total, three steps have to be fulfilled: extraction of chemical formulae from documents and further indexing, and the design of suitable rankings. They propose machine learning techniques based on Support Vector Machines (SVM) and Conditional Random Fields (CRF) for extracting chemical formulae. Furthermore, a rule-based string pattern match is introduced to improve the overall performance. However, chemists rarely use chemical formulae for information gathering due to their ambiguity.

Therefore, for the internal digital representation and exchange of structures several other text-based formats have been developed. Based on the algorithms developed by Morgan [23] and Gluck [24] it is possible to store two-dimensional atom-bond structural representations of chemical entities in a tabular form, so-called connection tables. In addition, linear notations have found widespread use. The early Wiswesser line notation (WLN) [25], or the later SMILES [26], ROSDAL [27] and SYBYL line notation [28] are representations of chemical structures in the form of

[12] http://bioie.ldc.upenn.edu

[13] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA

a linear string of alphanumeric symbols. The latest development is the InChI Code, an open standard for chemical structure description, by the IUPAC [29]. Nevertheless, the automatic extraction of these representations is a challenging task.

In chemistry, just a few open source chemical entity recognition tools are currently available. Prime examples are the OSCAR framework [6] and ChemSpot [8]. Both can identify and extract multiple name variations of chemical entities. OSCAR also uses name-to-structure algorithms to transform the found chemical entities into chemical structure information [30]. The recognition rates of OSCAR depend on the query term and have been evaluated varying between 69% to 81% for recall and 64% to 75% for precision [6]. A comparison of OSCAR and ChemSpot on the SCAI corpus [9] showed the following results [8]: the highest precision with 76.6% is reached by ChemSpot compared to 66% for OSCAR. But, the best recall of 82% is reached by OSCAR compared 71.9% by ChemSpot. Overall, both frameworks are useful to automatically extract chemical entities.

In the course of this thesis, we use the OSCAR framework to annotate all chemical entities contained within a document. These annotations are collected in a so-called standoff annotation file (annotated SciXML) which contains pointers to the respective elements in the source text. Of course, the automated recognition of chemical entities is still dealing with the challenges of ambiguity. However, as we will see later, indexing with automatically extracted phrases can already provide sufficient retrieval quality for most documents. The following algorithm summarizes the different steps to convert documents in SciXML and annotate the contained chemical entities.

1. */* Adjustment of algorithm parameters*/*
   Given a set of PDF documents define a corresponding set of regular expressions defining layout specific parameters, e.g., position of captions and table formats.
2. */* Convert PDF documents to their respective representation in HTML.*/*
   For each document do
   2.1. Convert to HTML using *pdftohtml*; this produces a HTML file for each page. The HTML encapsulates every coherent text fragment into a <DIV> element enriched by style descriptions like font size, font family and absolute position.
   2.2. Concatenate all pages of each document to a single file.
3. */* Removing unnecessary text fragments*/*
   For each HTML file do
   3.1. */* Calculate average line distance and length*/*
   Iterate over all <DIV> elements and determine the average line distance / length in paragraphs.
   3.2. */* Remove reference section **
   Identify the beginning of the reference section using the corresponding regular expression. Remove all succeeding <DIV> containers.

3.3. */* Remove tables */*
Identify all table captions using the corresponding regular expression. According to the general layout iterate over the succeeding (or preceding) <DIV> elements. Derive distances between each two elements using the position information. Once the distance is larger than the average calculated in 3.1 or a page break occurs, delete all <DIV> containers between the caption and the current position.

3.4. / Remove figures */*
While figures are already removed during the OCR process, text fragments contained in certain figures (e.g. chemical reaction schemes) may still remain. Therefore, identify all figure captions using the corresponding regular expression and remove all captions. Identify remaining text fragments: if the line length in any <DIV> container is shorter than the average, delete the respective element.

3.5. / Identify abstract and keywords */*
Identify the abstract / keyword section with the corresponding regular expression. Mark up the section as abstract / keyword by adding the respective class attribute to the <DIV> element.

3.6. / Convert SUB and SUP */*
Identify all candidates for sub- and superscript elements based on the absolute positioning and the font size. Convert the corresponding <DIV> element into a <sub> or <sup> element.

3.7. */* Merge paragraphs */*
Merge all remaining unclassified <DIV> elements into one single paragraph representing the document's fulltext.

3.8. */* Convert and save as SciXML */*
Convert the resulting HTML file into its corresponding SciXML representation using the SciXML Java Object Model[14].

4. / Entity extraction*/*
For each SciXML document do

4.1. Process all text with the OSCAR framework. This produces an annotated SciXML file marking up chemical entities, reactions, concepts and techniques.

Algorithm 1. Document conversion and entity extraction

2.2. Generating Enriched Index Pages

In the last section, we presented an algorithm converting documents into SciXML representations. All contained chemical entities have been automatically annotated

[14] http://www.l3s.de/vifachem/resources.html

using the OSCAR framework. However, these annotated files still only contain exactly the name and representation of the chemical entity found in the document. Thus, we also have to solve the problems of synonyms and different entity representations, like, e.g., SMILES or InChI codes. In general, we have two possible options to solve this problem. Either we extend the query term with synonyms and all other entity representations, or we enrich the documents with the related metadata. We decide to use the latter option due to the following reasons. The documents contained in our chemical digital library should also be found using textual searches, e.g., provided by Google or Yahoo!. Since most of the documents are located in the so-called deep Web, they cannot be indexed directly using Web crawlers due to underlying copyright restrictions. Furthermore, we do not have any influence on the query expansion methods used by Web providers. Thus, we decided to extend the documents with suitable metadata and created enriched index pages.

Use Case: The following scenario is typical for the daily work of a practitioner in the chemical domain. Assume our scientist is interested in the synthesis of odorous substances, e.g., as ingredients for perfumes. In particular, our chemist may be looking for building blocks usable in various synthetic pathways. Here, a simple precursor is the molecule *methoxybenzene* (see **Fig. 2** left), which is a common intermediate in the production of pharmaceuticals or odorous substances. In fact, a derivate of *methoxybenzene*, *1-methoxy-4-(1-propenyl)-benzene*, is the main component of anise oil (see **Fig. 2** right) which can be isolated by steam distillation from star anise (*Illicium verum*) or anise (*Pimpinella anisum*).

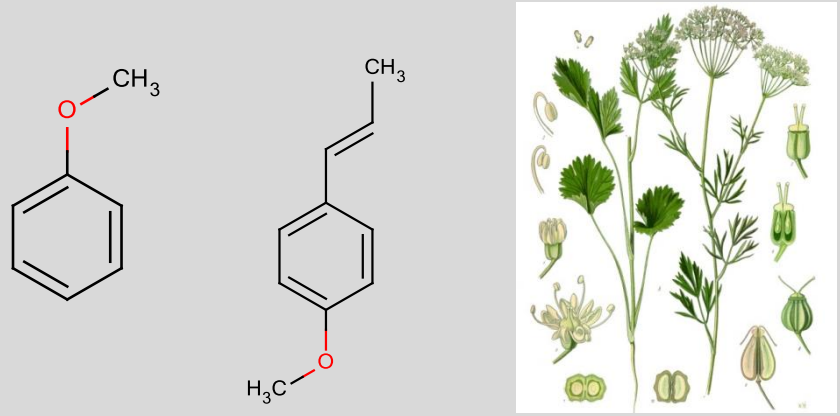


Fig. 2. Methoxybenzene and 1-methoxy-4-(1-propenyl)benzene (left)
Anise, from Koehler's Medicinal-Plants 1887 (right)

For the sake of open access, assume that in his/her search for information our practitioner lacks access to commercially available chemical structure databases (due to the high prices or license limitations). Focusing on a name-based search our practitioner has to face the challenge of disambiguating chemical names (IUPAC, INN, trivial or brand name). Picking up our example entity *methoxybenzene*, one could also search for *phenoxymethane*, *phenyl methyl ether*, or even the

trivial name *anisole*. All these names represent a valid verbal description of the substance. Therefore, our chemist first tries a keyword-based Web search using the query term '*methoxybenzene*', specifically on information from freely available open access journals.

For example, the ARKIVOC Journal is one of the oldest open access journals in Organic Chemistry, published since 2000, containing detailed experimental information about various compounds. But, for the ARKIVOC collection a search for '*methoxybenzene*' returns zero hits. Still, only given the fulltexts it is impossible to distinguish whether the document collection simply does not contain any document with the entity or if our practitioner has only selected a verbal descriptor of the compound not used within the documents. In fact, a query on '*anisole*' would have retrieved 7 correct results. Thus, providing and maintaining a proper index linking all relevant information about substances to the papers they occur in, is vital.

Moreover, practitioners, as well as academic researchers, are usually interested in finding all related documents to individual chemical entities. For both user groups the search is basically recall-oriented, because especially for synthesis procedures or production processes missing information about, for instance, existing patents or expected yields may lead to considerable financial losses.

Recently, a first few approaches trying to index digital chemical collections for keyword-style Web search have been proposed. For example, [19] and [20] both present naïve approaches to enable chemical search by indexing the empirical formulas of occurring substances. The basic assumption of their work is that chemists search for literature using a chemical formula. Since chemical formulas are ambiguous this is not the case in reality. Nevertheless, the formula search is integrated in the authors' search platform ChemXSeer [31], [17]. Harvard's QueryChem Portal[15] allows searching the Web based on an expanded query automatically generated from any chemical structure drawn in a graphical user interface [32]. First, the chemical structure is converted into a SMILES code, which in turn is used for a reference lookup in chemical Web databases like PubChem, ChemBank or Zinc. The lookup provides corresponding synonyms, which are then used for a Web search via the Google API. Although such a query expansion definitely is a first step, this approach can only rely on data already correctly indexed by Google. Since most chemical documents are hidden in chemical digital libraries, they still are not retrieved, even by an expanded query. Hence, the key to solve this problem lies in proper indexing.

To create our enriched index pages, we rely on the annotated SciXML format generated by OSCAR. The next step is to collect further metadata like synonyms, SMILES and InChI for all extracted chemical entities. Generally, this information can be retrieved from topic-centered databases. The most comprehensive open access

[15] http://www.querychem.com

database for the area of chemistry is PubChem[16]. However, for large scale metadata generation lookups using the PubChem Web interface or Web service is far too slow. To address this problem we used the PubChem SQL dump to store all entity data in a file based hash map. By using a random access file, it is now possible to directly access the relevant metadata, using the chemical name as key, without sequentially scanning the file. In fact, we measured a performance improvement of about two orders of magnitude in comparison to a Web service call: the hash map lookup needs for all kind of queries only 0.01 seconds in contrast to the Web service calls needing between 1.7 and 3 seconds depending on the complexity of the query. We also tried loading the PubChem dump into a relational MySQL database, which, however, still resulted in around 0.2 seconds response time for all queries.

In addition to the PubChem metadata, we also detect the role of a chemical entity in the respective document. This role describes the semantic meaning of a chemical entity in a document. Possible roles we defined are: product, reactand, catalyst, and solvent. We analyzed our chemical document collection and manually identified 36 often used lexico-syntactic patterns. The patterns for the identification of products are shown in Table 1, for a complete overview see Appendix A. These patterns are in pseudo code, meaning that *[CHEMICAL]* is a placeholder for a recognized named entity, marked up by OSCAR. The placeholders are numbered consecutively and replaced by the respective role.

Table 1. Lexico-syntactic pattern for the role identification of products

Taken Role	Lexico-syntactic pattern in pseudo code
PRODUCT	(?i).* Synthesis of (\s+[-\w|\p{InGreek}]*\s*){0,3} [CHEMICAL]
PRODUCT	(?i).* was used to prepare.* [CHEMICAL]
PRODUCT	(?i).* Giving \s [CHEMICAL]
PRODUCT	(?i).* Formation of \s [CHEMICAL]
PRODUCT	(?i).* One-pot synthesis of \s [CHEMICAL]
PRODUCT	(?i).* Preparation of (\s+[-\w|\p{InGreek}]*\s*){0,2} [CHEMICAL]
PRODUCT	(?i).* Yielding \s [CHEMICAL]
PRODUCT	(?i).* Leading to \s [CHEMICAL]
PRODUCT	(?i).* To afford (\s+[-\w|\p{InGreek}]*\s*) [CHEMICAL]
PRODUCT	[CHEMICAL] (?i).* were obtained from.*
PRODUCT	(?i).* To obtain (\s+[-\w|\p{InGreek}]*\s*){0,2} [CHEMICAL]

[16] http://pubchem.ncbi.nlm.nih.gov

The collection of generated index pages is now ready to be used as an extended search index over the documents collection. The beauty of our workflow is that the index pages can also be indexed and subsequently be retrieved by general purpose Web search engines, like, e.g., Google or Yahoo!. Most of the PDF presentations of the original documents are stored in the deep Web and therefore have not been indexed before. Using HTML based index pages they are found by the users and can be accessed while meeting the copyright restrictions of the publishers. Furthermore, since the pages also include SMILES representations for most entities, it is also possible to pose structural queries by entering the SMILES code of the query entity in a Web interface. The following algorithm summarizes the different steps we performed to generate enriched index pages.

1. */* Enrich chemical entity metadata.*/*
   For each standoff annotation file do
   1.1. Create a corresponding index page in HTML.
      1.1.1. Fill the header's <TITLE> element with the journal name and paper title.
      1.1.2. Adding available <META> fields out of the Dublin Core Metadata Element Set into the header container.
      1.1.3. Add the paper's title within a <H1> tag.
      1.1.4. Copy the paper's abstract into a paragraph.
      1.1.5. Link the index page to the original URL.
      1.1.6. Create an empty table for the enriched entity metadata.
   1.2. For each chemical entity marked up in the standoff annotation file do
      1.2.1. Use the PubChem hash map to retrieve all corresponding metadata.
      1.2.2. Detect the role of the chemical entity using the lexico-syntactic patterns
   1.3. Add a table row storing the chemical entity with all metadata and its role.

Algorithm 2. Generation of enriched index pages

2.3. Evaluating the Quality: Comparing Enriched Index Pages and Structure Search

For our evaluation, we used a collection of 2588 chemical documents from the journal *Archive for Organic Chemistry* (ARKIVOC)[17], which is one of the most renowned open access sources for organic chemistry. For each document, we created an enriched index page as described in the previous section. To assess the difference between a Web search over our semantically enriched index pages and plain fulltext retrieval we used a simple Lucene whitespace analyzer to build an inverted index for the fulltext documents (baseline) and the enriched index pages. For structure search

[17] www.arkat-usa.org

the chemical entities are stored in a MySQL database in a structure table constructed by ChemAxon[18].

Basically, we performed four different experiments:

- First, we evaluated the impact of our enriched index pages in terms of average result set relevance. The results of randomly chosen text queries were evaluated in a precision/recall analysis.
- To evaluate the quality in terms of ambiguity resolution we compared the retrieval results using enriched index pages to an exact structure search.
- To show the practical applicability of our approach especially over large document collections we also compared the respective retrieval times of structure and text search.
- Since our global aim is to expose chemical document collections hidden in digital libraries via commonly used Web search interfaces, like, e.g., provided by Google or Yahoo!, we made our enriched index pages available online. Then we analyzed the number of pages crawled by Google and to what degree our pages are actually indexed.

2.3.1. Impact of Enriched Index Pages

In this experiment, we evaluate the impact of our enriched index pages using a precision/recall analysis. Relevance can only be assessed manually by domain experts, in what is a very expensive process. Therefore, we performed the precision/recall analysis only on a subset of documents (still about 10% of the entire collection). To choose a *representative* subset, we analyzed the number of occurrences of individual chemical entities in the document collection. **Fig. 3** shows the distribution of the 5000 most often occurring chemical entities.

Since it is not sensible to choose entities for evaluation that either occur in almost all documents or are extremely rare, we chose our query entities for evaluation only from entities occurring in less than 100, but more than 20 documents. We retrieved all documents matching the queries and randomly chose a subset of 10%. From these documents, we randomly selected a total of 5% of the occurring entities resulting in 22 textual query terms varying from trivial entity names to InChI codes. For the evaluation domain experts considered all retrieved documents with respect to each query and judged the relevance in a binary fashion.

To determine the practical value of our textual indexing, the domain experts used a very strict relevance rating: documents are only marked as relevant, if there was an exact match for the query entity regarding both syntax *and* semantics. For example, the relevance judgment distinguished between actual substances and substance

[18] www.chemaxon.com

classes. Since classes are often simply given in the plural form of the respective substance this poses a difficulty for stemming in text search engines. Even worse, in some documents complex entities are described using a basic entity name as placeholder for a more detailed structure shown in some image. Since the actual structure may have totally different chemical properties also such documents have been considered as errors in the relevance analysis. Finally, sometimes an entity name can even be used as a placeholder for describing certain characteristics or functionality of other entities, i.e. although some entity name may occur in a paper, the actual entity may not be relevant. The experts also counted such documents as false retrievals in the text search.

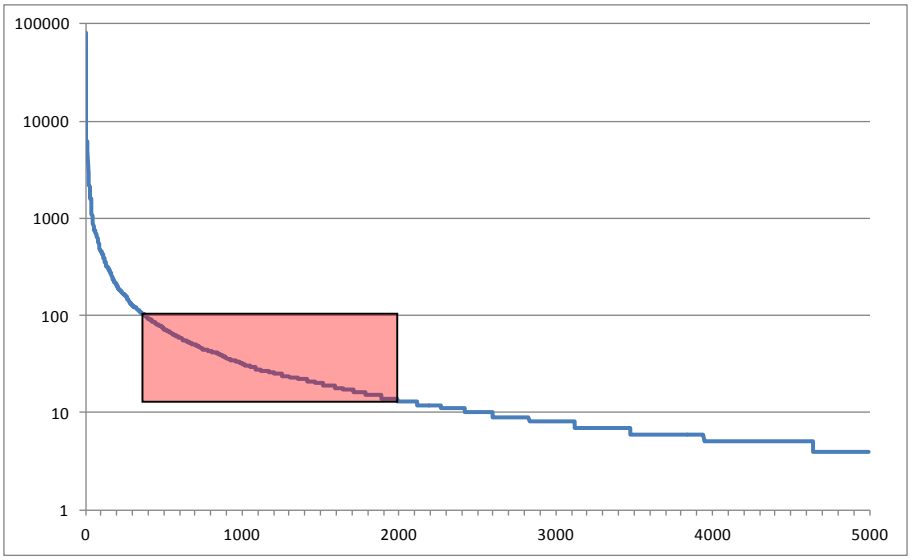


Fig. 3. Distribution of entity occurrence in documents

In total from all documents retrieved as query results the domain experts marked 158 documents as relevant regarding the respective queries. **Table 2** shows the resulting precision/recall values.

Table 2. Precision and recall values for baseline and enriched search

Search type	Retrieved	Retrieved + Relevant	Recall	Precision
Baseline	87	58	0.3671	0.6667
Enriched	259	150	0.9494	0.5792

As expected, we experienced a very low recall value of only 36.71% for the baseline approach. In contrast, the recall for our enriched index pages is 94.94%. The semantic enrichment thus yields essential benefits. For example, there will almost never be a hit in the baseline fulltext documents for queries on InChI codes, whereas our index pages include the InChIs and all synonyms of the query term for most of

the structures. But, given the strict relevance voting necessary for practical usefulness this tremendous recall benefits have to be paid for in terms of precision. Still, the precision of our approach has only slightly decreased at 57.92% compared to 66.67% for the baseline documents. Basically, due to our enrichments the result set size grows, however, this increases also the number of technically correctly found, but semantically irrelevant documents.

Table 3. F_x-Measure values for baseline and enriched search

Search Type	F_1-Measure	F_2-Measure	$F_{0.5}$-Measure
Baseline	0.4735	0.4033	0.5731
Enriched	0.7194	0.8418	0.6281

To also quantify the overall benefit of our enrichment technique we computed the weighted F-Measures. **Table 3** shows the different F-Measure values of the different search types. For the classic F_1-Measure we can see already a dramatic improvement of more than 0.2 over the baseline. Moreover, document retrieval in the area of chemistry is rather recall oriented: it is very important to retrieve *all* documents related to query. For an industrial research team missing relevant research results (e.g., with respect to patents) may lead to enormous costs for the respective company. Hence, the actually most significant measure for our scenario is the F_2-Measure weighing recall higher than precision. Here our algorithm even scores an improvement of more than 0.4. But even when a user focuses on a precision-oriented search, our algorithm still results in a small benefit of 0.05 for the $F_{0.5}$-Measure.

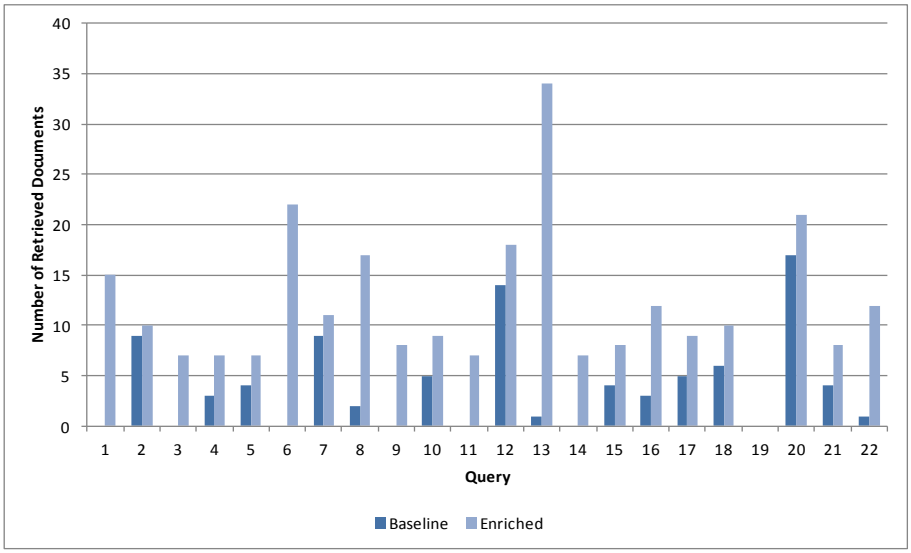


Fig. 4. Retrieved documents per query: enriched versus baseline search

Investigating the search results per query more closely we found that the benefit can really be seen in all searches. **Fig. 4** shows a detailed overview of the number

of retrieved documents per query. For all queries the enriched index pages retrieved more relevant documents than the baseline search. An exception is query 19 where no matching document was found in either approach. The respective query term *InChI=1S/C5H8O/c1-2-4-6-5-3-1/h2,4H,1,3,5H2* cannot be found because the responsible entity in the original document could not be matched uniquely to the PubChem entities. As we can see, there is still need for further improvement for metadata enrichment.

2.3.2. Quality of Enriched Index Pages

To measure the quality of our enriched search approach we compared the results to a chemical structure search, which currently is state of the art for chemical digital libraries. However, a structure search has complex requirements: it is necessary to use specialized commercial software, e.g., ChemAxon's JChem suite, to build up a structure database. The structural data is stored in a proprietary format (varying dependent on the vendor) and also the access to the data is only possible by using appropriate graphical query interfaces where structures can be sketched.

Structure search applications offer different query types: beside an exact structure search also sub-/super-structure and similarity searches are possible. Unfortunately, these search types are not directly portable to textual searches, because, e.g., substructures of an entity are not simply substrings of the entity name. Therefore, we have to focus on exact matching structures in our experiments, and leave other kinds of structure searches to future work. For each of our query terms we took the corresponding structure information of the chemical entity and retrieved all matching documents.

Table 4. Precision and recall values for enriched and structure search

Search type	**Retrieved**	**Retrieved + Relevant**	**Recall**	**Precision**
Enriched	259	150	0.9494	0.5792
Structure	262	154	0.9747	0.5878

Table 4 shows that the recall value for our enriched index pages of 94.94% is very similar to the respective value for the structure search. And also the precision values of 57.92 % for enriched and 58.78% for structure search are almost identical. Hence, also the F-Measures shown in **Table 5** are nearly the same. Please note, that although structure search has more complex requirements, it offers only a slight advantage for exact matching queries over searching our enriched index pages.

Again, we investigated this effect on query level. **Fig. 5** compares the retrieved documents for each query entity. As expected from the precision/recall analysis, in most queries enriched and structure search retrieved the same number of documents.

Table 5. F_x-Measure values for enriched and structure search

Search Type	F_1-Measure	F_2-Measure	$F_{0.5}$-Measure
Enriched	0.7194	0.8418	0.6281
Structure	0.7333	0.8613	0.6385

The only exceptions occur for queries 12, 13 and 19. We already commented on the ambiguous entity term in query 19; of course a structure search can resolve this ambiguity accounting for the slightly increased recall of structure search. Moreover, for queries 12 and 13 some irrelevant documents were found in the text search, because the query entity was a substring of some more complex entity occurring in the document. For example, the query term for query 12 is *iodobenzene*. Here, also irrelevant documents containing entities, like, e.g., *diacetoxyiodobenzene* or *tetraiodobenzene,* are retrieved. Also the abbreviated naming of entities by using their functional groups only contributes to the false retrievals.

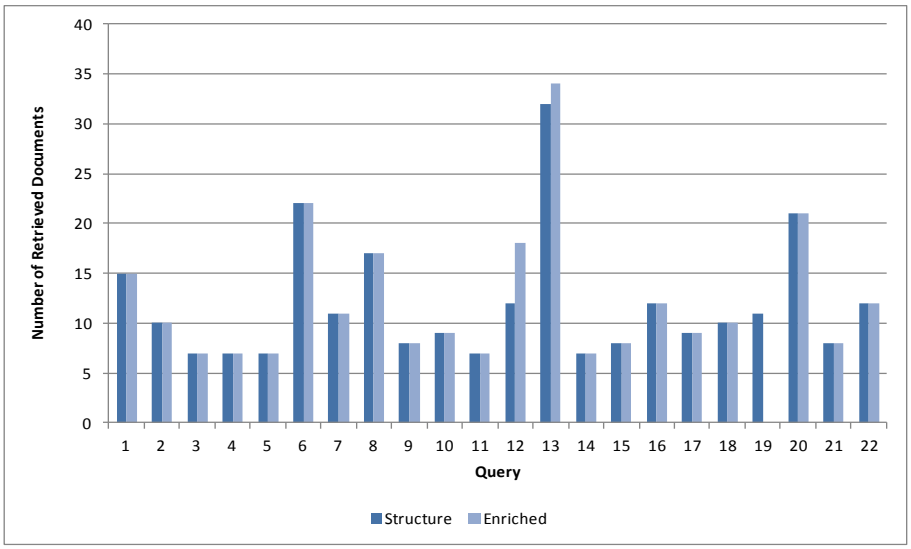


Fig. 5. Retrieved documents per query: enriched versus structure search

To summarize this experiment, we can state that a text search on enriched index pages indeed yields similar results to a chemical exact structure search with respect to the retrieved documents.

2.3.3. Search Performance

In this experiment, we compare the respective retrieval performance in terms of response times for text- and structure search. The measured time comprises query processing until all relevant documents have been retrieved. We performed experiments over several days on our digital library server to get representative average values. We did three batches, each run including 10.000 queries, varying the query terms for the text search between SMILES, names and InChI codes. The 10.000

query entities were chosen randomly from our entity database. For the structure search always the SMILES code is used which is internally converted into a unique structure representation of the respective entity. Please note, that usually also the drawing of the actual structure followed by a conversion into a SMILES code or CML would be part of the structure search. We discounted these costs by directly starting from the SMILES code. In any case, the conversion of linear notations to fingerprints is a step that has always to be performed in structure search independently of whether a SMILES code is directly given or the structure is drawn. After finding the exact matching entity for that structure all related documents are retrieved.

In text searches beside single term queries also query terms concatenated with Boolean operators are commonly used. Therefore, we simulated 'AND' and 'OR' searches. Since in structure search Boolean queries are not easy to perform, the only way here is to make two subsequent structure searches. **Fig. 6** shows the average retrieval times measured for the different search types.

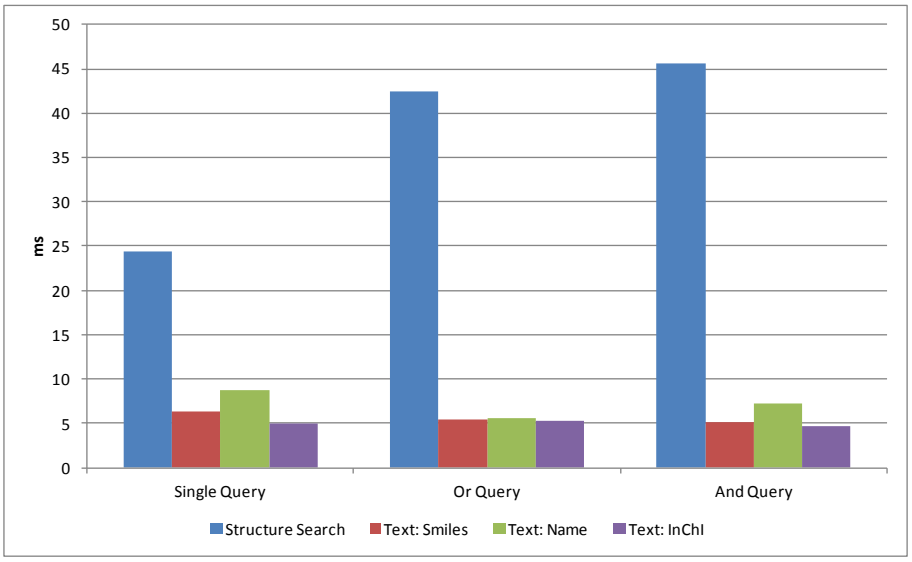


Fig. 6. Retrieval times [ms] for different search types

As a general trend, we can see that text searches are far more efficient. For instance, in text search, it makes no difference which query term is used or if more than one term is concatenated in a Boolean query. The retrieval times only vary between five and eight milliseconds (note that name search is slightly less efficient than SMILES or InChI, because of many synonyms). Using a structure search the document retrieval is always about an order of magnitude slower due to the complex matching of fingerprints. Moreover, the time for queries using Boolean operators is rather high, since here two (or more) structure searches are needed (in our experiments we only used simple queries comprising two terms).

In summary, our results show that a text search is always much faster than a structure search independently of the text search's query term. Moreover, for Boolean queries the retrieval time for text queries does not increase.

2.3.4. Indexing for Web Search

Our overall aim is to improve access to chemical document collections hidden in digital libraries via common Web search providers. Therefore, we simply made all enriched index pages for the ARKIVOC journal available on the Web. To have a chance of being indexed the generation and layout of our enriched pages is important. Most crawlers would mark pages within a site as spam, if they just show some index terms and do not include at least some fulltext or links. Therefore, our pages include, beside the actual enriched metadata table, the document's title, its abstract and a link to the fulltext. On the other hand, high quality open access journals will also feature high PageRanks, thus crawlers will index them prominently.

After three month of being online, the Google index indeed contained already around 600 of our pages. However, it is not traceable how the pages are indexed and exactly why a page is indexed and some other not. **Fig. 7** shows a screenshot of a text search on the term '*InChI=1S/C5H8O2/c1-3-5(6)7-4-2/h3H,1,4H2,2H3*'.

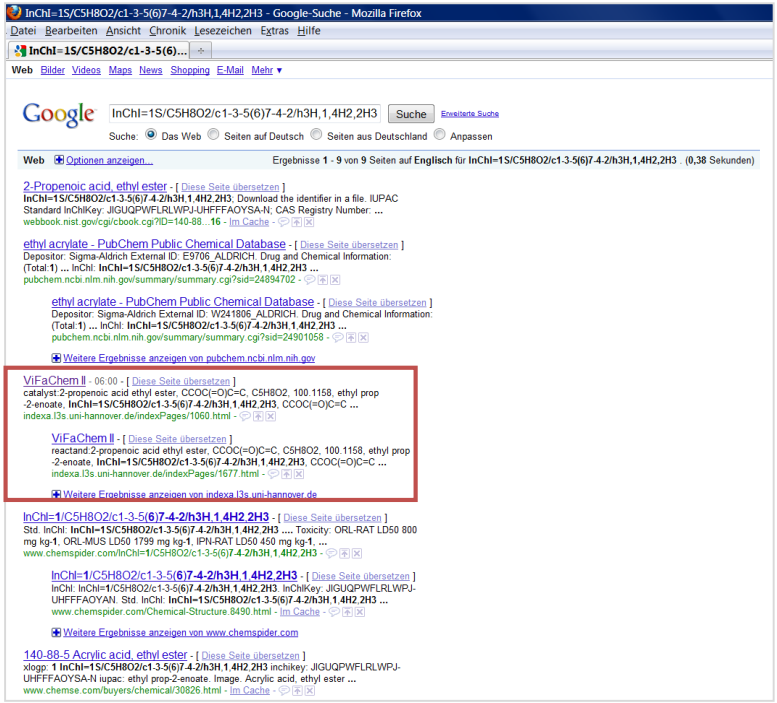


Fig. 7. Google search example for InChI code

The enriched index page for the relevant ARKIVOC journal paper '*Effect of substituents and benzyne generating bases on the orientation to and reactivity of haloarynes*' appears on third place in the Google result, directly after the respective dictionary

entries of the substance from the National Institute of Standards and Technology and the PubChem substance database.

Although we did nothing to promote the index pages, i.e. our pages still have a Google PageRank of zero (as opposed to PageRank seven for both NIST and PubChem), they are still found and provide access to relevant documents that would not have been found otherwise (as the respective ARKIVOC journal papers do never appear in the Google search result). Please note that for investigating the indexing process we always chose '*ViFaChem II*' as title for all of our enriched pages to detect them easily in the Web search results. Of course, usually the journal name and title of the related document is used.

2.4. Conclusions

In this chapter, we have shown how to enable text-based searches in chemistry. For each document, an enriched index page was created, containing different entity representations, like, e.g., SMILES and InChI codes, as well as synonyms for each chemical entity from the respective document. Using these index pages we were able to open up chemical literature hidden in digital libraries and enable text queries in commonly used search interfaces, like, for instance, provided by Google or Yahoo!. Our experiments have shown the usefulness of our approach. The retrieval quality of our enriched index pages is almost as good as chemical exact structure searches and significantly better compared to a baseline/fulltext search.

Chapter 3

Similarity Search

In the last chapter, we have built the basis to enable text-based queries by creating enriched index pages containing all synonyms, different entity representations and roles. The index pages also contain structural information for most chemical entities, namely the SMILES code. Therefore, we are able to use structural information using a basic textual query interface, instead of complex graphical interfaces. Using these index pages we are now able to do exact matching textual queries. The simplest workflow for a retrieval system allowing for such queries is shown in **Fig. 8**.

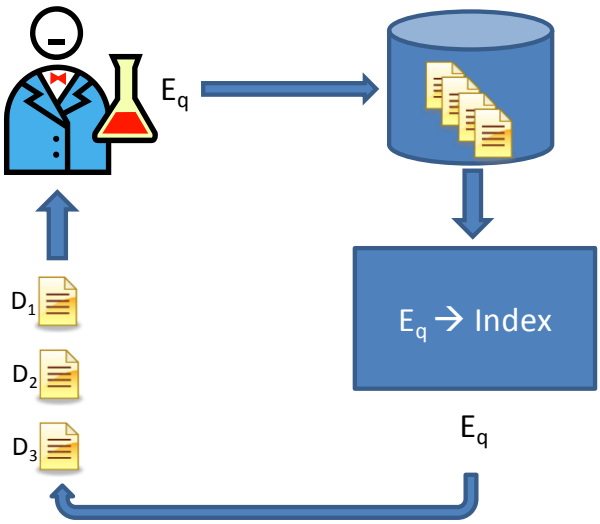


Fig. 8. Simple workflow

The user submits his query entity E_q to the search engine. Searching for relevant documents regarding E_q is difficult since we have to take all different entity representations (e.g. SMILES or InChI codes) and synonyms into account. To address ambiguity, we rely on our chemical index pages to search for relevant documents. Note that, due to the fact that in chemical documents the most relevant entity, i.e. the product of a synthesis, can occur only once, only Boolean queries are reasonable and traditional IR measure, e.g., TF*IDF, are not.

Use Case: Imagine a chemist from the field of drug design who is currently working on an improvement of Viagra®. He/she is especially searching for related literature about the active ingredient *Sildenafil* (see **Fig. 9**).

Our chemist can use the simple architecture and search for documents about '*Sildenafil*'. Unfortunately, he is still unable to fulfill his information need, because the active pharmaceutical ingredient '*Sildenafil*' is trademarked and cannot be used

for other drugs. As a consequence, he must relax his query to find other chemical entities with similar properties. Indeed, this query relaxation should be done automatically by replacing the actual entity with similar entities.

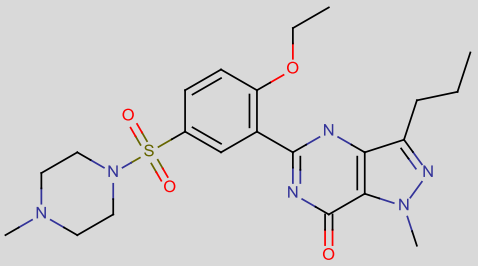


Fig. 9. Structure of Sildenafil

In this chapter, we show how to perform similarity searches in chemistry. However, until now, using the index pages, it is only possible to pose exact matching queries. But, as described in the use case it is important to have the ability to find entities that are similar to the query entity. In chemistry, there are many different similarity measures based on structural information, i.e., the SMILES code in our case. All of them have in common that they rely on unique fingerprint representations of the chemical entities. Thus, the first necessary step for computing similarity is the transformation of a chemical substance into a fingerprint.

3.1. Fingerprint-Based Similarity Measures

Fingerprints encode molecular structures in a series of binary digits (bits) where bits are set according to occurrences of particular structural features. For generating fingerprints, we use the SMILES representation [26] stored on the index pages. There are several ways of creating fingerprints focusing on different fragments of chemical entities. Examples for typical fragments for generating fingerprints are:

- *Atom sequence*: A linear path of atoms and bonds through the molecule.
- *Ring composition*: An atom and bond sequence around a ring structure in the molecule.
- *Atom pairs*: A pair of atoms in the same molecule with number of bonds in the shortest path between them. The different atom pairs are usually further differentiated by, e.g., taking the number of attached hydrogens into account.

Sometimes fragments are too specific, leading to very low frequencies and sparse fingerprints. This results in similarity values that are not meaningful to distinguish chemical entities. In the course of this thesis we rely on the open source chemical development toolkit (CDK) [33], [34], which includes the following fingerprints.

Standard Fingerprint This fingerprint examines the molecule and encodes the following:

- a pattern for each atom

- a pattern representing each atom and its nearest neighbors
- a pattern representing each group of atoms and bonds connected by paths up to 2 bonds long
- a pattern representing the atoms and bonds connected by paths up to 3 bonds long
- a pattern representing the atoms and bonds connected by paths up to 4, 5, 6, and 7 bonds long

Extended Fingerprint An Extended fingerprint includes in addition to the Standard fingerprint features for describing aromatic rings.

Graphonly Fingerprint This fingerprint is a specialized version of the Standard fingerprint that does not take the bond order into account.

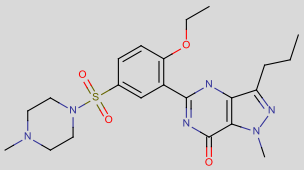
EState fingerprint generates 79 bit fingerprints using fragments describing the electronic and topological characterization of an atom, called electrotopological state (e-state) [35]. The fingerprint simply indicates if such a fragment is present in the structure or not.

MACCS Fingerprint is the representation of the answer of 166 questions about a chemical structure [36].

Substructure Fingerprint currently supports 307 different substructures. A set bit indicates that the related substructure was found in the molecule.

Example: Substructure Fingerprint generation

Let us consider a chemist who is searching for *Sildenafil*. In a first step the name is converted to its unique SMILES representation: *CCCC1=NN(C2=C1NC(=NC2=O)C3=C(C=CC(=C3)S(=O)(=O)N4CCN(CC4)C)OCC)C*. This conversion is necessary, because SMILES codes include information about the molecular structure of a chemical substance. Now we want to create a Substructure fingerprint out of this SMILES code. For simplicity, let us consider that the Substructure fingerprint takes only four substructures into account. Each of the substructures is encoded in a SMARTS[19] pattern:



1. C=N-N-C: Pattern for an atomic arrangement taking bond orders into account.
2. C-S: Pattern for an atomic arrangement taking bond orders into account.
3. N-Br: Pattern testing the existence of a N-Br bond.
4. Oc1ccc(O)cc1: Pattern testing the presence of a specific substructure.

For each matching SMARTS pattern, we set the corresponding bit to 1. The resulting fingerprint for Sildenafil is 1100.

[19] http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html

Since a lot of fingerprint transformations are available, the amount of possible combinations of fingerprints and similarity computations between them is really high. The idea of measuring the similarity of two objects, each defined by a set of common attributes, is discussed in many different domains, including, e.g., biology [37] or chemistry [13]. Although these application areas are divers, the used similarity coefficients are almost the same. Since the performance always relies on the choice of an appropriate measure, many researchers have worked on finding the most meaningful measure. The work done by Willet et.al, see [13] and [38], gives overviews of the coefficients that have found widespread use in chemical information systems.

Even though numerous binary similarity measures have been described in the literature by their properties and features [39], [40], [41], [42], only a few comparative studies are available. In the field of biology, Hubalek collected 43 similarity measures and after evaluating similarities, correlations, transformations of the value range and symmetry, 23 were excluded. The remaining ones were used for cluster analysis on fungi data to produce five clusters of related coefficients [37]. In the domain of chemistry, Willet evaluated 13 similarity measures [43]. All of these measures rely on a unique fingerprint representation of the chemical structure. Considering these fingerprints, we examined the most common useful measures (see **Table 6**) in the domain of chemistry collected in [38]. The variables of the formulas are defined as follows: If we consider two fingerprints of two chemical entities A and B, then:

- *a* is the count of bits set to 1 in entity A but not in entity B
- *b* is the count of bits set to 1 in entity B but not in entity A
- c is the count of the bits set to 1 in both entity A and entity B
- *d* is the count of the bits set to 0 in both entity A and entity B

Table 6. Reviewed similarity measures

Measure	Range	Formula
Cosine	[0, 1]	$\frac{c}{\sqrt{(a+c)*(b+c)}}$
Dice	[0, 1]	$\frac{2*c}{(a+c)*(b+c)}$
Euclidean	[0, 1]	$\sqrt{\frac{c+d}{a+b+c+d}}$
Forbes	[0, ∞]	$\frac{c*(a+b+c+d)}{(a+c)*(b+c)}$
Hamman	[-1, 1]	$\frac{(c+d)-(a+b)}{a+b+c+d}$
Jaccard / Tanimoto	[0, 1]	$\frac{c}{a+b+c}$
Kulczynski	[0, 1]	$0.5*\left(\frac{c}{a+c}+\frac{c}{b+c}\right)$

Measure	Range	Formula
Manhattan	[1, 0]	$\frac{a+b}{a+b+c+d}$
Matching	[0, 1]	$\frac{c+d}{a+b+c+d}$
Pearson	[-1, 1]	$\frac{(c*d)-(a*b)}{\sqrt{(a+c)*(b+c)*(a+d)*(b+d)}}$
Rogers-Tanimoto	[0, 1]	$\frac{c+d}{(a+b)+(a+b+c+d)}$
Russell-Rao	[0, 1]	$\frac{c}{a+b+c+d}$
Simpson	[0, 1]	$\frac{c}{\min((a+c),(b+c))}$
Tversky	[0, 1]	$\frac{c}{\alpha*a+\beta*b+c}$
Yule	[-1, 1]	$\frac{(c*d)-(a*b)}{(c*d)+(a*b)}$

In order to better understand the differences between these similarity measures, we first examined to which degree the similarity measures are correlated.

3.1.1. Correlation Analysis

Since now, there is no work done in the literature, analyzing the correlation of the similarity measures applied on different fingerprints. Thus, our first goal was to explore if the underlying fingerprint has some influence on the similarity measures. To do our first experiment, we took a random 1‰ sample of the PubChem database resulting in around 48.000 chemical entities. We downloaded their SDF files to have the structural information of all entities and converted them into their respective SMILES representations. These SMILES codes were necessary to generate the different fingerprint representations of each chemical entity using the CDK. In addition, we randomly choose 20 chemical entities as query entities. Since, in a later step, we want to use the similarity measures for retrieval it seems reasonable to evaluate not only the complete result set of around 44000 entities but also smaller subsets. Thus, we decided to also evaluate the differences between the top-k results. Therefore, we computed for each combination of fingerprint, chemical entity and top-k the 16 fingerprint based similarity measures resulting in around 88 million similarity values.

As we can interpret the similarity value as a value in a ranking vector, we decided to use the Kendall rank correlation coefficient (KTau) [44] to determine the correlation of the different measures and fingerprints. We calculated the correlation coefficient for each ranking vector and the arithmetic mean over 20 queries. A KTau of 1 means that the agreement of two rankings is perfect, -1 indicates a perfect disagreement and for independent rankings one would expect the coefficient to be

approximately 0. Our experimental results have shown that the actual KTau values strongly differ over the fingerprints. For example, the KTau value for the combination 'Euclidean / Russell-Rao / EState fingerprint' and 'Euclidean / Russell-Rao / Standard fingerprint' varies from 0.53 to -0.30 (see **Table 7**).

Table 7. Similarity measures with highest variances over EState (1), Extended (2), Standard (3), Graphonly (4), MACCSS (5) and Substructure (6) fingerprint

Similarity Measure	1	2	3	4	5	6
Tanimoto / Euclidean	0,83	0,12	0,11	0,39	0,67	0,76
Cosine / Matching	0,82	0,05	0,04	0,40	0,67	0,76
Dice / Rogers Tanimoto	0,83	0,12	0,11	0,39	0,67	0,76
Euclidean / Russell-Rao	0,53	-0,29	-0,30	-0,09	0,38	0,33
Manhattan / Russell-Rao	-0,53	0,29	0,30	0,09	-0,38	-0,33
Tversky / Forbes	0,48	-0,11	-0,09	0,23	0,17	0,54
Forbes / Kulczynski	0,39	-0,40	-0,35	0,14	0,04	0,41
Hamman / Russell-Rao	0,53	-0,29	-0,30	-0,10	0,37	0,32
Jaccard / Rogers Tanimoto	0,83	0,12	0,11	0,39	0,67	0,76
Kulczynski / Euclidean	0,83	0,00	0,01	0,43	0,68	0,76
Matching / Russell-Rao	0,53	-0,29	-0,30	-0,09	0,38	0,33
Pearson / Russell-Rao	0,73	0,10	0,11	0,33	0,60	0,59
Rogers Tanimoto / Russell-Rao	0,53	-0,29	-0,30	-0,09	0,38	0,33
Russell-Rao / Rogers Tanimoto	0,53	-0,29	-0,30	-0,09	0,38	0,33
Simpson / Euclidean	0,66	-0,17	-0,11	0,32	0,48	0,55
Yule / Russell-Rao	0,67	0,01	0,02	0,19	0,50	0,49

Due to the definition of the KTau, it is not straightforward to depict the uncorrelated similarity measures because *approximately zero* is not a well-defined threshold. To ensure a relatively high likelihood of correlation, we defined a threshold of 0.8. Based on this threshold, we evaluated how many uncorrelated similarity measures we have for each fingerprint. The results are shown in **Fig. 10**.

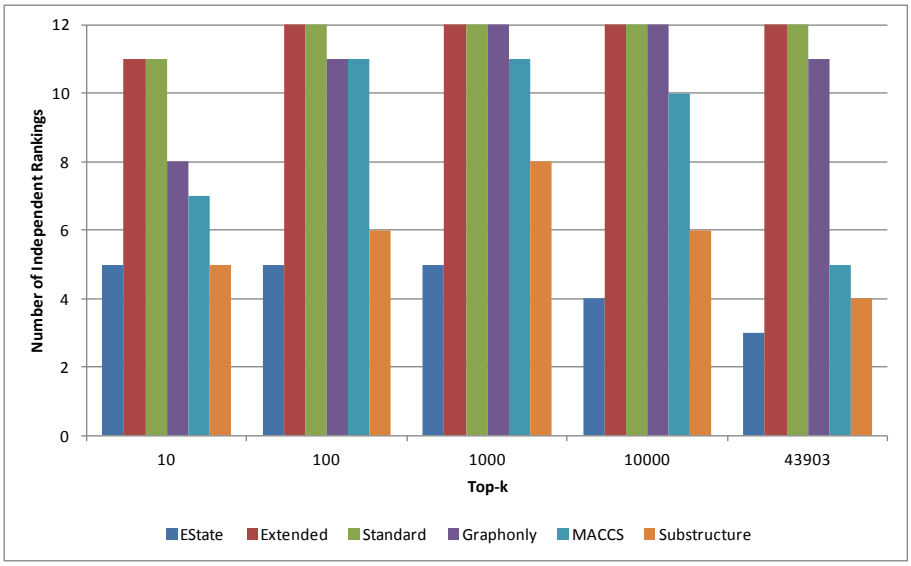


Fig. 10. Number of minimal independent rankings for top-x and a threshold of 0.8

Interestingly, the EState fingerprint always has the minimum number of uncorrelated similarity measures. The reason might be that the Estate fingerprint is only 79 bits long and, therefore, has less discriminative power than, for example, the substructure fingerprint with 309 bits. Still, also for the Estate fingerprint the concrete number of uncorrelated measures differs from five to three, which means that we have to take at least three different similarity measures (i.e. Yule, Russell-Rao and Forbes) into account. Given this result, we noticed that taking only the correlation coefficient into account is not discriminative enough; thus we consider additional discriminative properties.

3.1.2. Task-Based Analysis

This huge variety of uncorrelated similarity measures is eligible, because chemical similarity differs according to the task a chemist is working on. Intuitively, we considered that each measure might be useful for a specific task and, therefore, conducted experiments with example tasks using synthesis and drug design. For drug design, we took, among others, *Sildenafil* as query entity. The idea is to retrieve alternative substances with similar chemical properties. In Chapter 4, we extend the idea of the search task to the more general idea of search context. Whereas we define the task as the working field of the chemist, the search context is much more

precise, like, e.g., searching for similar entities to *Sildenafil* regarding the side effect of *irregular heartbeat*.

Use Case: Let us consider there are two scientists from the area of drug design Peter and Bob. Both are searching for *Sildenafil*, but with different additional conditions. Peter is interested in *pyrazolopyrimidinones* with a piperazine ring system connected to the sulfonyl group. In contrast to Sildenafil, Peter is looking for a free N-side at the piperazine to examine further reactions at this position. A good hit for this query scenario is *Demethylsildenafil* (see **Fig. 11**).

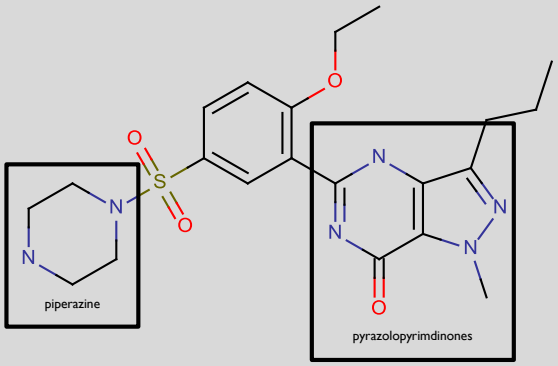


Fig. 11. Demethylsildenafil

Bob is interested in *pyrazolopyrimidinones* with a secondary amine connected to the sulfonyl group, as he is interested to perform alkylation reactions at his position. *Udenalfil* with its N-alkylated secondary amine side chain represents a top candidate for this kind of query (see **Fig. 12**).

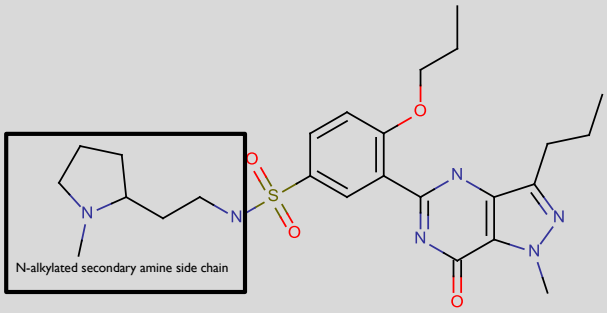


Fig. 12. Udenafil

To evaluate the ranking results of the different similarity measures, we took all chemical entities that were retrieved by a similarity search in PubChem for the query term *Sildenafil*. We also assured that the entities of interest defined by the domain experts, *Demethylsildenafil* and *Udenafil*, are included in this set. We computed similarity values for *Sildenafil* and each entity in this set using all uncorrelated similarity measures. The domain experts analyzed all result sets and evaluated which similarity measure retrieves the best ranking. The output of the experiment is that there is no suitable measure delivering both as relevant defined entities

under the top-10. For Peter who expected *Demethylsildenafil* as relevant hit the combination of EState fingerprint and Tanimoto measure delivers the best results, ranking *Demethylsildenafil* on rank 9 and *Udenafil* on rank 335. For Bob expecting *Udenafil* as most relevant entity the combination of Substructure fingerprint and Tanimoto measure gives the best result, ranking *Udenafil* on rank 2 and *Demethylsildenafil* on rank 228. Although both chemists are from the field of drug design, they expect different rankings for the same query term. Therefore, it is not possible to use one fixed similarity measure for one specific task. Of course, we also tried queries for the other tasks, but with the same result: it is not possible to assign one similarity measure to a specific task.

To better judge the impact of the task, we interviewed a group of domain experts to find reasons for this behavior. We figured out that each individual chemist has some kind of special background or implicit knowledge that he implies. We define this implicit knowledge as everything that is influencing the subjective notion of relevance of the chemist, like, e.g., costs for synthesis or which substances are already in the fund of the company. This background knowledge cannot be expressed by the query term resulting in insufficient result sets.

One possible solution is to build a personalized retrieval system. The idea is that each individual user trains the system and the system will learn the similarity measure, which fits best to his needs. **Fig. 13** shows our advanced workflow.

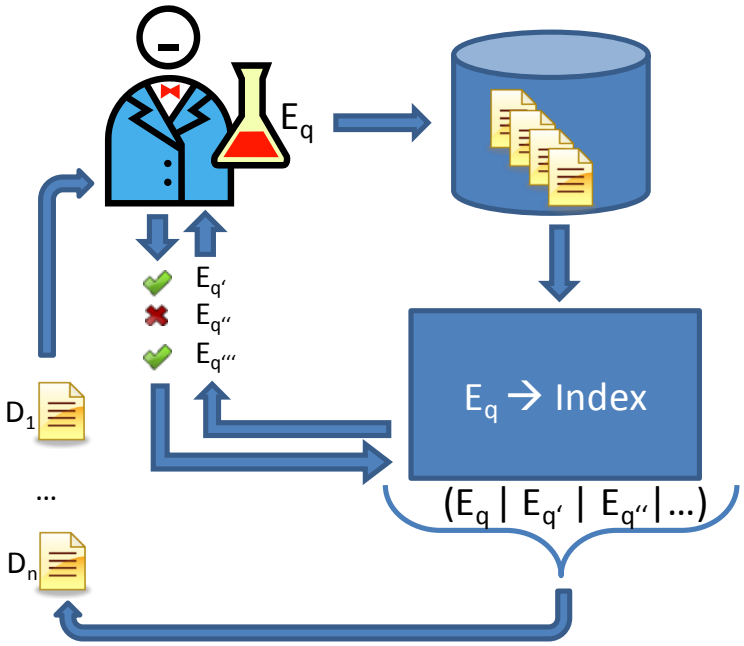


Fig. 13. Advanced workflow

In addition to the simple workflow, we add a query relaxation module to be able to relax the query E_q with similar entities. The result set only includes documents containing the chemical substance E_q or at least one other similar entity for example

$E_{q'}$. The document result set is ranked according to the similarity value of the included entities. As a result of the ranking function, the documents containing E_q are always top ranked followed by documents including the most similar entity $E_{q'}$.

A lot of uncorrelated measures are available resulting in totally different rankings and it is not obvious which similarity measure / fingerprint combination is most applicable. Therefore, our system contains a feedback step where each user marks chemical entities most relevant for his query. For a new user, the system uses the *best* similarity measure by computing the arithmetic mean over all available user feedbacks and learns the best individual similarity measure in some feedback steps.

For each feedback step, the system is calculating the top-k results of all uncorrelated measures for a query. Out of this list, the user has to decide which chemical entities are relevant for him. In the next step, the system calculates the precision at 10 values for each measure and uses the best matching one. If the chosen measure does not change over a number of different queries, it is accepted as default measure for this user and the feedback step is skipped for subsequent queries. Of course, if the user is not satisfied by the proposed ranking, he can force the system to learn or to use another measure. To evaluate the system, we conducted a user study with domain experts from the area of drug design and synthesis. We want to discover if already such a simple feedback step would result in an explicit combination of similarity measure and fingerprint. Furthermore, we are interested in the number of feedback cycles that are necessary until such a system is stable.

For the user study, we have randomly chosen 10 query entities from PubChem, each of them representing one feedback cycle inside the system. Based on the results shown in the previous chapter, we used the five uncorrelated measures Russell-Rao, Yule, Forbes, Simpson and Manhattan for calculating the similarity values. In the first step, we retrieved the top-10 entities for each similarity measure and put them in one set, which did not include duplicates and was unranked. In the second step, the chemists marked all relevant entities resulting in their personalized ranking vector. For each query, we took the respective ranking vector and compared it to the top-10 vector of the uncorrelated similarity measures by computing precision at 10.

Example: As an illustrating example we take the results of the domain expert introduced in our use case scenario searching for *Sildenafil* (see **Fig. 14**). One can see that there are perfect candidates for the personalized similarity measure, i.e. a combination of the Extended fingerprint and the Yule, Forbes or Simpson measure. However, of course one query is not enough to decide for a specific similarity measure.

Fig. 15 shows the average precision at 10 values for the chemist regarding 10 different queries. Regarding all queries, the personalized similarity measure has slightly changed. Finally, the best matching similarity measure is Russell-Rao based on the Graphonly fingerprint. Only six feedback cycles were necessary to find this ideal combination for this chemist, meaning the preferred similarity measure did

not change again after 6 queries. The second best measure is the combination of Yule and the Extended fingerprint.

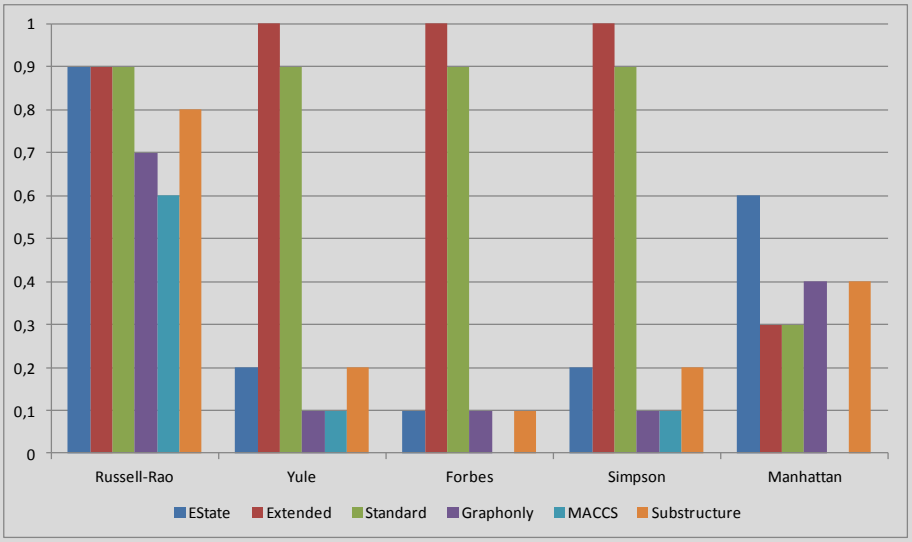


Fig. 14. P@10 values for the query *Sindenafil*

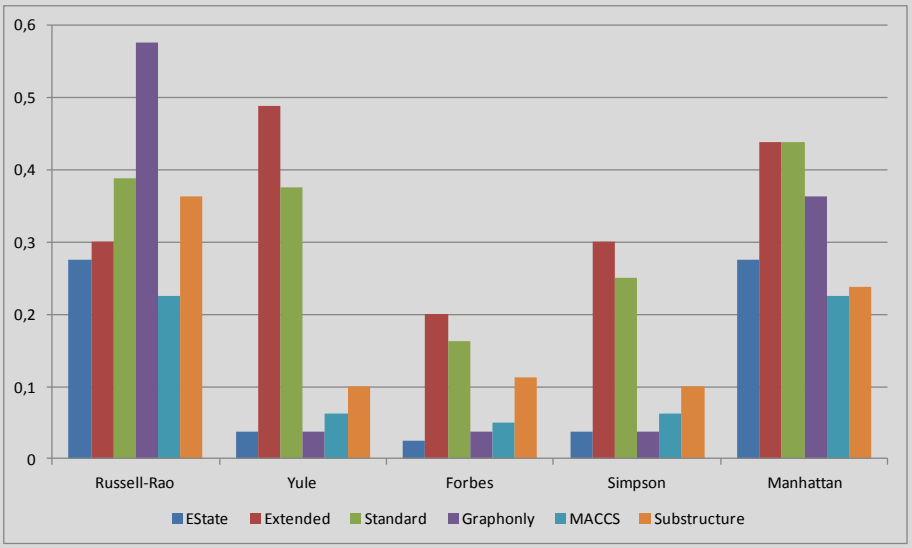


Fig. 15. Average P@10-values for one chemist over all queries

The second question to evaluate was the number of needed feedback cycles until the system was stable for an individual user. For this purpose, we defined the system as stable, if the precision value did not change more than 2% over three queries. We can state, that for 75% of the domain experts, the system was able to determine an explicit combination of similarity measure and fingerprint within our ten feedback cycles. The particular number of needed feedback cycles varies between three and eight. For the remaining 25% we could not determine a combination after 10 feedback cycles.

Furthermore, we analyzed the arithmetic mean over all experts and queries (see **Fig. 16**). One can see that the Russell-Rao measure outperforms all other measure applying it on the EState, Graphonly, MACCS and Substructure fingerprint. The best measure for the Extended fingerprint is Yule and for the Standard fingerprint it is Manhattan. Remember, these results cannot be applied out of the box to all users because the individual expectations can differ a lot. However, they are candidates for solving the well-known *new user problem*, if the user decides at least on a specific fingerprint or taking the overall best measure for a global starting point, i.e. the combination of Russell-Rao and the Graphonly fingerprint.

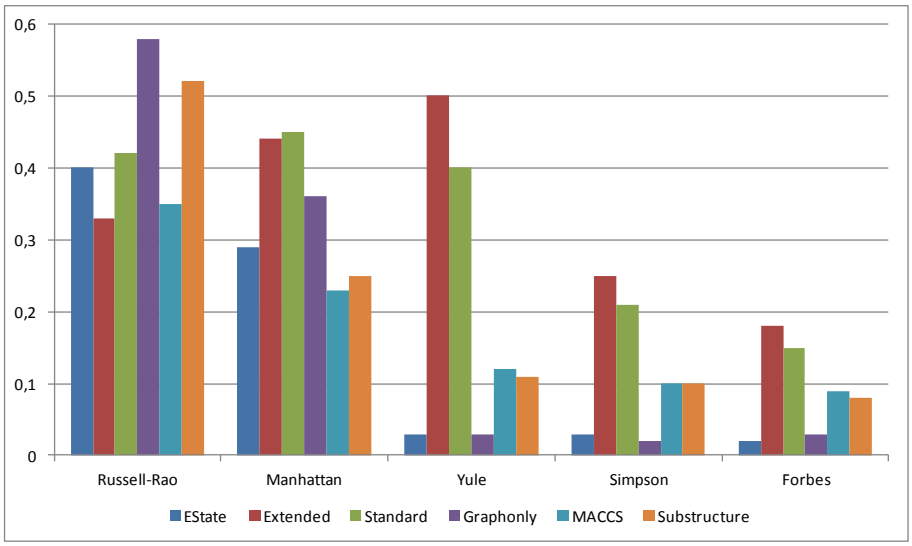


Fig. 16. P@10 values for arithmetic mean over all experts and queries

Although the personalized workflow already gives better retrieval results, we are interested in modeling the chemists' implicit knowledge to further improve the retrieval quality.

3.2. Similarity Considering Implicit Knowledge

In this section, we show a way to model the implicit knowledge of a chemist. Again, we focus on the important field of drug design, where the information gathering and indexing process is even more complex. A chemist from this area is not only interested in a specific chemical entity, but in a representative of a chemical class adopting a specific role. Especially, he is interested in entities having the same or similar characteristic chemical reactions. To assess if a chemical entity is relevant for his task in mind he uses his implicit knowledge about chemical classes and reaction behaviors. The characteristic reaction behavior of a chemical entity is defined by its functional groups. Functional groups are specific groups of atoms that will undergo the same or similar chemical reactions independent of the molecule they are part of. However, currently there is no knowledge base available allowing for this kind of automatic classification of chemical entities.

Use Case: Assume a scientist from the area of drug design who is interested in anti-tuberculosis drugs, their pharmacological activities and synthesis. He may start by looking for information about *Isoniazid* and related drugs. *Isoniazid* is an organic compound and the treatment of choice for tuberculosis (see **Fig. 17**). Thus, it is of high interest for pharmaceutical research since its discovery in the 1950s and its first synthetization in the early 20th century. Naturally, our researcher is looking for experimental procedures for the synthesis of Isoniazid-like structures, as he would like to minimize the side effects and the risk of resistance. In a first step, the chemist analyzes the structure of *Isoniazid* and identifies the parts of the molecule responsible for the specific reaction behavior. Furthermore, he implicitly knows that *Isoniazid* belongs to the chemical class of *hydrazines*. In particular, he is interested in chemical substances having the same reaction behavior and chemical class as *Isoniazid*.

Fig. 17. Structure of Isoniazid (left) the treatment of choice for tuberculosis (tubercle bacillus) (right)

As starting point, our chemist will use *4-cynaopyridine* (see **Fig. 18**) as it is already available in his laboratory. The question to solve is how to synthesize *4-cynaopyridine* to get a substance having the same functional properties as *Isoniazid*.

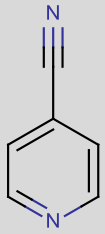


Fig. 18. 4-cyanopyridine

Therefore, our chemist is searching for literature where the synthesis of *Isoniazid*-like structures is described. Furthermore, also the chemical entity *4-cynopyridine* should be included as educt. The first step is the search for entities with the same functional groups as *Isoniazid*: *hydrazine derivative, aromatic compound, carboxylic acid hydrazide, heterocyclic compound*. Furthermore, relevant entities must also belong to the class of *hydrazines*. Finally, the result set is filtered for papers including the chemical entity *4-cynopyridine* as an educt. As final step, the chemist

can now examine, if the reaction described in the papers can also be used for his chemical entity.

3.2.1. Calculation of Functional Groups

Focusing on organic chemistry and especially on the synthesis of chemical entities the most important characteristic for retrieving relevant entities regarding a query term are the functional groups. We rely on the command line utility *checkmol*[20] to determine the functional groups of a chemical entity. Checkmol analyzes the input molecule for the presence of approximately 200 functional groups. We analyzed the output in a first short experiment with a group of domain experts and find out that checkmol simply recognize the presence of an *aromatic ring*, but does not further investigate the dimension of contained aromatic rings. To enhance the quality of the resulting clusters, we added an extra parsing step to checkmol's output, to determine the dimension of an aromatic ring, resulting in n/m-aromatic rings, where n stands for the number of contained aromatic rings and m for the number of connected ring groups. An example is shown in **Fig. 19**. The left figure shows the chemical structure of *Phenanthrene* which is a (3/1) aromatic compound. The structure contains three aromatic rings combined to one group. In **Fig. 19** (right) the chemical graph of *Dicumarol* is shown which is (2/2) aromatic compound containing two aromatic rings partitioned into two groups.

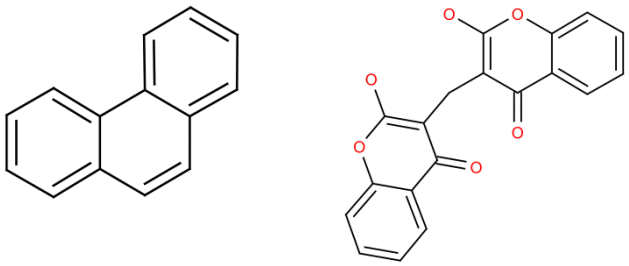


Fig. 19. Phenanthrene is a (3/1) aromatic compound (left)
Dicumarol is a (2/2) aromatic compound (right)

3.2.2. Clustering Based on Functional Groups

The goal is to build clusters of chemical entities where each entity is located in one cluster and all other entities in that cluster have the same functional groups. We used a dump of the PubChem database containing around 31.5 million chemical entities. For each entity, we determined the functional groups and created an inverted index with name and entity allocation. In the first experiment, we want to gain first insights about the entities contained in the PubChem dump. We downloaded all SDF

[20] http://merian.pch.univie.ac.at/~nhaider/cheminf/cmmm.html

files from the PubChem server and extracted each chemical entity. A SDF file contains a lot of different information about the entity itself. For us, the interesting part contains information about the chemical structure. Therefore, we extracted the entity's SMILES code and used our extended version of the command-line tool *checkmol* to determine the respective functional groups, resulting in a set of functional groups for each entity. All entities containing exactly the same set of functional groups are grouped into one cluster. The cluster label is a MD5-hash, which is computed using the concatenation of all functional group names from that cluster. The distribution of all 31.5 million entities is shown in **Table 8.**

Table 8. Cluster sizes

# Contained Entities	# Clusters
1	773092
1 < x ≤ 10	816817
10 < x ≤ 100	226147
100 < x ≤ 1000	36535
1000 < x ≤ 10000	3615
10000 < x ≤ 100000	143
100000 < x	0

We did a survey with domain experts to analyze the clusters. The result is that clusters containing up to 100 chemical entities are still reasonable for domain experts meaning they correspond to the chemist's implicit knowledge.

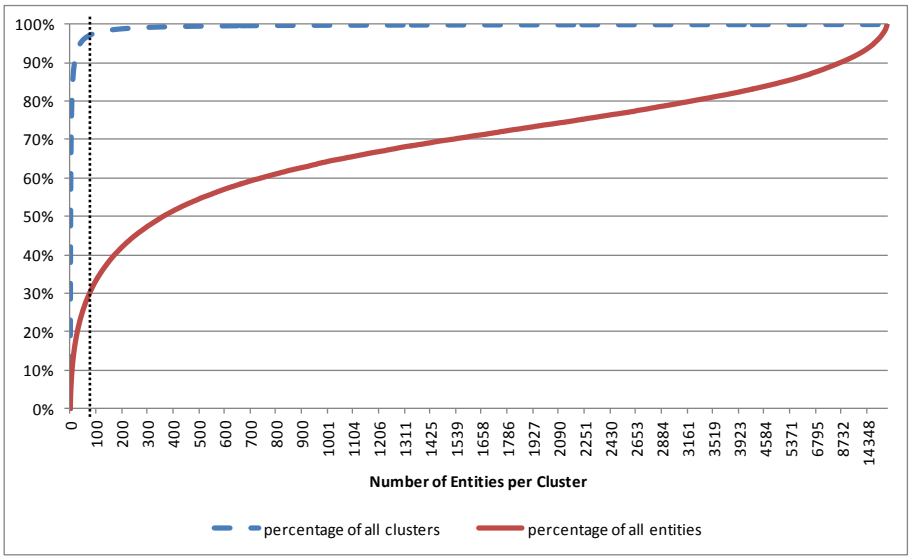


Fig. 20. Number of entities per cluster

Therefore, we can discover that 97.84% of the resulting clusters can already be used. But these clusters only contain around 30% of all chemical entities (see **Fig. 20**). Most of the entities (around 21 million) are located in the remaining 2.16% of the clusters. Therefore, it is necessary to split them up into more meaningful clusters.

3.2.3. Building Meaningful Sub-Clusters

Since the clusters containing the majority of all chemical entities contain entities from different chemical classes, we further decomposed them into sub-clusters by computing fingerprint-based similarity between the included entities. However, the clustering of objects always relies on a meaningful distance, respectively similarity function. For computing similarity between chemical entities, usually the first step is to create a fingerprint representation of the entity. In the next step, commonly known similarity measures are used based on these fingerprints. As previously shown there are a lot of different uncorrelated fingerprint/similarity measure combinations that need to be taken into account. Each of these measures has different rankings with respect to the underlying fingerprint. The week point of all measures is the complexity for calculating the similarity value. Assuming n is the number of entities, it is necessary to compute the similarity between each pair of entities resulting in a complexity of $O(n^2)$. By clustering entities with similar functional groups this complexity can be dramatically reduced. The pre-computation of the functional groups can be performed in constant time and thus results in a complexity of $O(n)$.

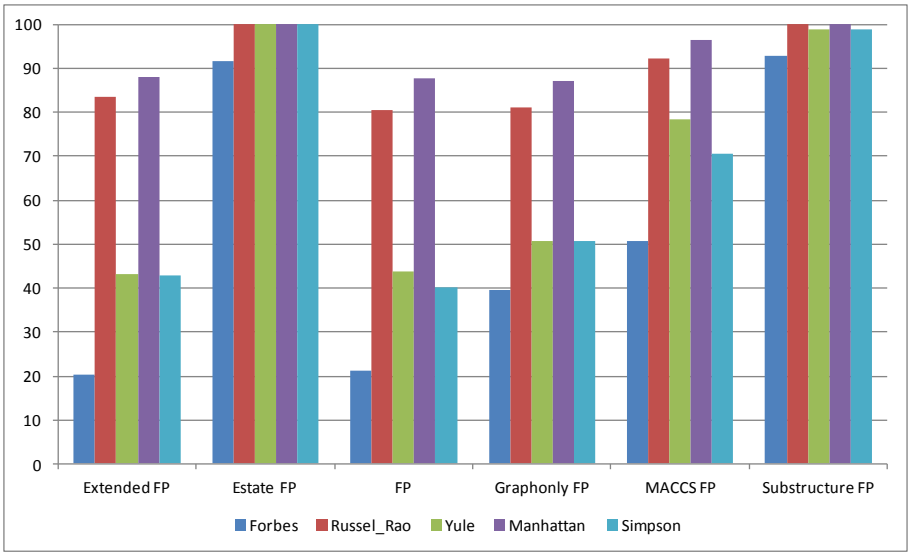


Fig. 21. Top-100

To decide for a measure for the sub-cluster computation, we evaluated for which of the measures the top-k ranked entities are in the same functional group cluster. We want to find the measure that best reflects the concept of functional similarity.

We randomly choose 100 clusters with more than 1000 entities per cluster. In addition, we choose 10 random queries and calculated the similarity between the query entity and all other entities from all clusters. For each entity, we have six different fingerprint representations. The similarity is computed using the uncorrelated measures. **Fig. 21** and **Fig. 22** show the average results based on the ten queries.

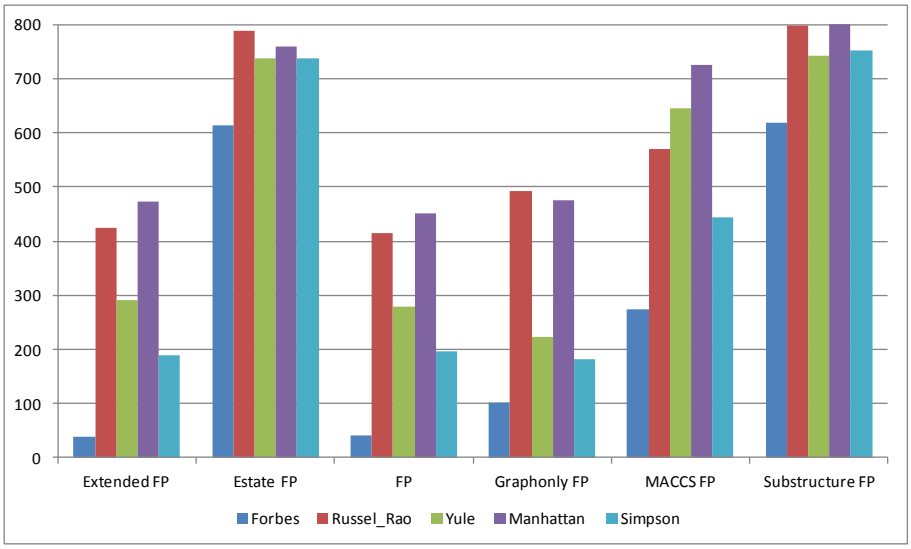


Fig. 22. Top-1000

The figures show that there are big differences between the different combinations. For the *substructure fingerprint* and the *Manhattan* distance always all top-100 entities are in the same functional groups cluster. For the top-1000 entities still around 800 are found in the same cluster. Since this combination retrieves the best results, we decided to use it for sub-cluster computation.

Many different clustering algorithms exist in literature. For our problem definition, we choose a partitioning method which constructs k partitions of the data. Each partition represents a cluster and satisfies the following requirements: each group must contain at least one object and each object must belong to exactly one group. One of the most famous algorithms from this group is the k-means clustering, which we chose using the WEKA framework [45].

As already discussed, a lot in literature one challenging aspect of k-means clustering is to find a suitable k. We also tried to find an optimal k fitting for our scenario. The aim is that each entity in a cluster has the same chemical class. Therefore, we took a domain specific ontology including chemical classes as ground truth, the so-called ontology for chemical entities of biological interest (CheBI [46]). Since we are interested in decomposing the clusters with more than 100 entities (around 40000 clusters), we randomly took 2000 clusters (5%) out of this set. Since not all entities from our dataset are included in CheBI, we only choose clusters containing entities also included in the CheBI ontology. The idea is to take all entities from one cluster and assign the associated ontology classes to that cluster. Of course, it is not sensible

to use all ontology nodes associated with one chemical entity. Nodes that are too general would lead to huge clusters that are again not meaningful.

As we will see later in section 5.2, it is sensible to only use ontology nodes that are at least three steps away from the entry node. Therefore, we only associated these classes with the respective cluster. We defined that the optimal segmentation is achieved, if all entities with different classes are in different sub-clusters. We manually built the respective sub-clusters and run the k-means algorithm varying the value for k. Our algorithm stops if k-means found the optimal solution, each entity is in one cluster for its own, or if no solution can be found. Evaluating the 2000 clusters we retrieved an optimal k for further splitting up the entities in the functional groups clusters in chemical classes of four. Whereas CheBI includes for our dataset around 20,000 chemical classes, we were able to find more than 150,000 classes for chemical entities. Please note, we cannot associate exact chemical class names to each cluster, but as we will see later, our results match the perception of the chemist's implicit knowledge of entities belonging to the same class. The following paragraph gives an overview of the whole process.

1. /* *Calculation of functional groups* */
   For each entity from document do
   1.1. Use SMILES code of the entity to determine the functional groups using checkmol
   1.2. If functional groups contain aromatic ring
      1.2.1. Parse output to determine the dimension of the aromatic ring
   1.3. Name the cluster according to the MD5-hash of the set of functional group names
   1.4. If functional groups cluster already exists
      1.4.1. Assign entity to that cluster in the inverted index
   1.5. Else
      1.5.1. Create functional groups cluster and add entity to the inverted index
2. /* *Analyzing cluster for building sub-clusters* */
   For each cluster *c* from the set of all clusters *C* do
   2.1. If entities in *c* belong to different chemical classes
      2.1.1. For each entity e in *c* do
         2.1.1.1. Compute fingerprint representation for e
         2.1.1.2. Compute sub-clusters based on fingerprints and distance measure using k-means clustering
         2.1.1.3. Add e to the inverted index of the sub-cluster

Algorithm 3. Cluster computation

3.2.4. Confining the Result Set: Retrieval Using Implicit Knowledge

In this section we explain how the functional groups clusters are used for retrieval. After the query entity is assigned to the respective sub-cluster all related documents

are retrieved. Instead of just delivering all documents, the result set is ordered according to a similarity measure. In our scenario, a document is relevant for a query term if some chemical entity in the document has the same functional properties, respectively the same chemical class, as the query entity. Therefore, we need a specific similarity measure not only taking the simple occurrence of the query term into account. We developed a measure, which is based on the Wikipedia category information. Our experiments presented in Chapter 5.2 prove that Wikipedia categories are useful to describe chemical documents. The Wikipedia categories are structured in a taxonomic tree based on the relationships between them. Here, the idea is to retrieve for each document the associated categories based on the included chemical entities. Since Wikipedia includes information from many different domains, it is not sensible to use the whole category tree for describing chemical entities. Based on the findings in Chapter 5.2, we use only categories that are directly attached to the query node. We retrieve the respective categories for each query term and each document in the query's sub-cluster. The documents are ranked according to the following similarity measure:

$$wc(q_i, d_j) = \frac{cq_i d_j}{cq_i} \times \frac{cd_j}{ed_j} \qquad (1)$$

where q_i is the query term and d_j the respective document. The *swc* measure consists of two parts. The first quotient divides the number of categories found for query term *i* in the respective document *j* ($cq_i d_j$) by the total number of categories found for query term *i* (cq_i). The second quotient divides the total number of categories for the document (cd_j) by the total number of chemical entities found in that document (ed_j). The following algorithm summarizes all necessary steps.

1. */* Document retrieval */*
   For query entity *q* do
   1.1. If input format ≠ SMILES
      1.1.1. Get SMILES representation for *q*
   1.2. Use SMILES to compute functional groups for *q* using checkmol
   1.3. Associate *q* with the respective cluster
   1.4. If functional groups cluster is not divided into sub-clusters
      1.4.1. Retrieve all included documents using the inverted index and rank them according to our similarity measure
   1.5. Else
      1.5.1. Compute fingerprint representation of *q*
      1.5.2. Compute distance to the centroid of each sub-cluster
      1.5.3. Assign *q* to sub-cluster with lowest distance
      1.5.4. Retrieve all documents included in the sub-cluster using the inverted index and rank them according to our similarity measure

Algorithm 4. Cluster based document retrieval

To evaluate the created clusters in a document retrieval scenario, we use our ARKIVOC document collection containing 2700 documents. We took the enriched index pages and associate each document to the functional groups clusters based on its contained entities. First, we have to randomly chose query entities and assess the relevance of each document for the respective query. Relevance can only be assessed manually by domain experts (in particular chemists), in what is a very expensive process. Therefore, we could not take the entire collection, but chose a subset of documents (still about 10% of the entire collection) for performing a precision/recall analysis. To choose a *representative* subset, we analyzed the number of occurrences of individual chemical entities in the document collection. It is not sensible to choose entities as query terms that either occur in almost every document or are extremely rare. We analyzed all entities occurring in less than 100 documents, but more than 20 documents. Furthermore, the entities should belong to functional groups clusters, which have been further decomposed into sub-clusters using the fingerprint based similarity computations.

We retrieved all documents matching these queries and randomly chose a subset of 10%. From these documents, we randomly selected a total of around 5% of the occurring entities resulting in 18 textual query terms. For the evaluation domain experts from the field of chemistry considered all retrieved documents with respect to each query and judged the relevance in a binary fashion. A document is marked as relevant if it contains entities having the same reaction behavior and belonging to the same chemical class as the query entity. However, sometimes an entity name can even be used as a placeholder for describing certain characteristics or functionality of other complex entities, i.e. although some entity name may occur in a paper, the actual entity may not be relevant. The experts counted such documents as false retrievals.

Now, we analyze if the sub-cluster decomposition is sensible meaning that all relevant documents for a query term are located in the same sub-cluster. If we are using only the functional groups clusters (k=1) the recall is 93%, meaning that some documents from other clusters were also marked as relevant. But, without any sub-clusters we got a low precision value averaged over all queries of 42% (see **Fig. 23**). The goal is to find suitable sub-clusters restricting the number of retrieved documents resulting in high recall and better precision values. Therefore, we build sub-clusters using the substructure fingerprint representation of the chemical entities and computing the similarity using the Manhattan distance. Since we use a k-means algorithm, we analyze the sub-cluster quality dependent on k. **Fig. 23** shows the precision, recall and F-measure values for different values of k.

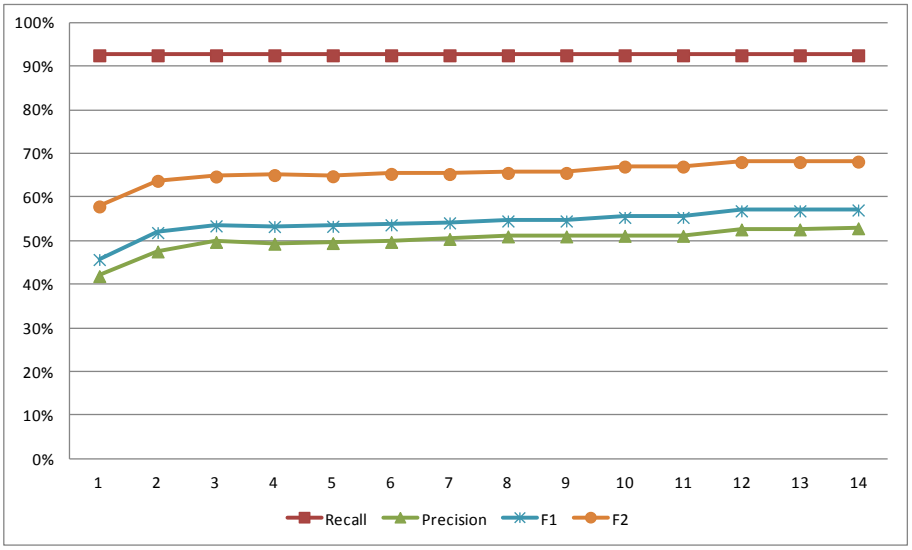


Fig. 23. Recall, Precision and F-Measures for varying k's

The recall value is always around 93%. The precision value slightly increases up to 53% for k equals 12. According to the low precision values the classic F_1-Measure is on average only around 57%. But, as stated before, document retrieval in the area of chemistry is rather recall oriented: it is very important to retrieve all documents related to a query. For an industrial research team missing relevant research results may lead to enormous costs for the respective company. Hence, the actually most significant measure for our scenario is the F2-Measure weighting recall higher than precision, resulting in an average of 68%.

Please note, we can further optimize the precision by using different k values for different queries, respectively different cluster sizes. For example, for query term 2, which is in a quite small cluster containing only 26 documents, already with k equals 2 we have a precision value of 60%. In contrast, for query term 15 the precision value for k equals 2 is 59% and for k equals 9 67%. Regarding all queries, the optimal value for k is varying between 1 and 12. But, only four queries do not have their optimal precision value for k equals 12.

The entity clustering experiment (see section 3.2.3) has shown that we have an optimal segmentation for k equals four. For documents, k equals four is already good, but the precision value is slightly higher for k equals 12. We also tested higher values for k, but the precision did not increase anymore. Instead of delivering all documents in the sub-cluster randomly to the user, we also developed a similarity measure to rank the documents according to the query term. **Fig. 24** shows the Mean Average Precision (MAP) values for varying k's. Using the ranking, we were able to reach a MAP value for k equals 12 of 72%.

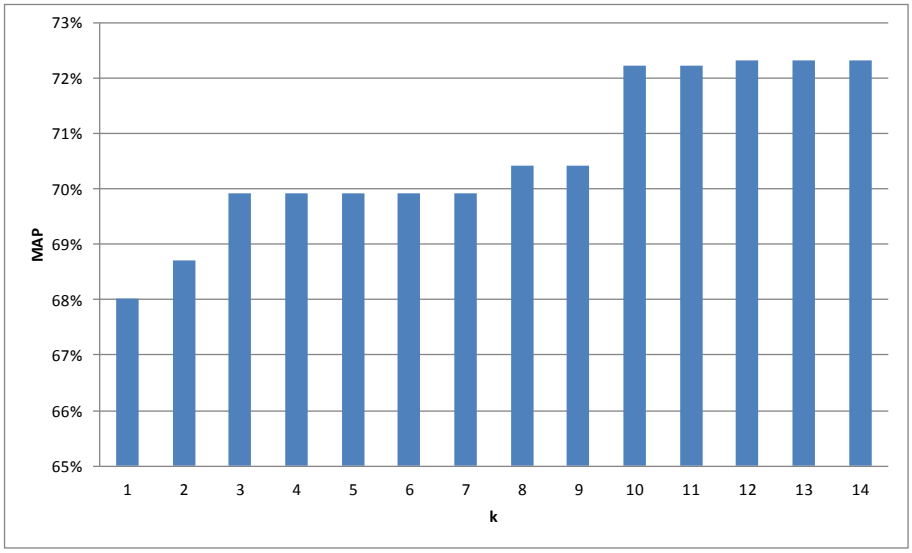
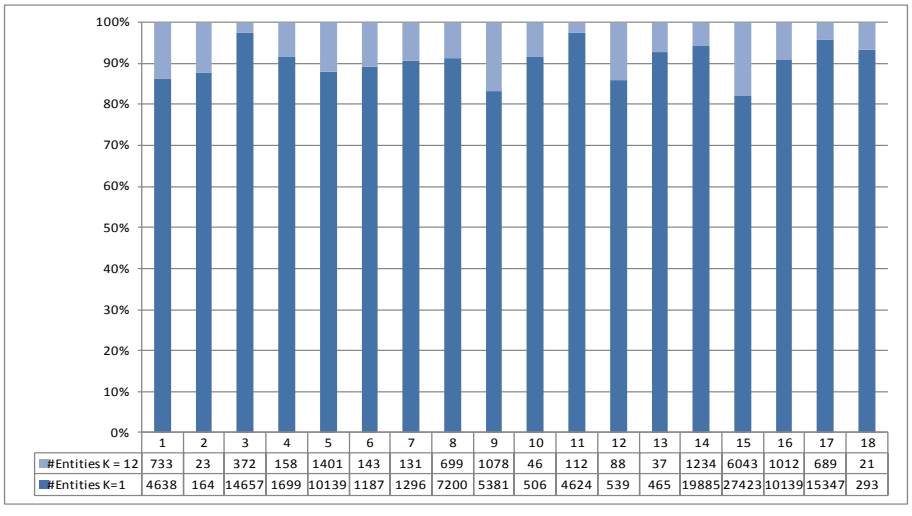


Fig. 24. Mean Average Precision (MAP) for Wikipedia categories ranking and varying k's

However, another interesting point is that even if the clusters include fewer documents, the recall value did not decrease. That means if a user is searching for documents with respect to a chemical entity with some characteristic reaction behavior and implicit knowledge of its chemical class, it is sufficient to find the cluster for this query entity and retrieve all included documents. If the query entity is located in a sub-cluster, it is not necessary to take chemical substances from other sub-clusters into account, even though they have the same functional groups. Our experiment has shown that almost all relevant documents are in the same sub-cluster as the query entity.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
#Entities K = 12	733	23	372	158	1401	143	131	699	1078	46	112	88	37	1234	6043	1012	689	21
#Entities K=1	4638	164	14657	1699	10139	1187	1296	7200	5381	506	4624	539	465	19885	27423	10139	15347	293

Fig. 25. Number of entities for k=1 and k=12

Fig. 25 compares the number of entities in the functional groups cluster (k=1) to the number of entities for 12 sub-clusters for each query. The figure shows that the

cluster sizes decrease on around 90% on average over all queries. Since the recall does not decrease, the conclusion is that the cluster quality is high and only irrelevant entities are located in other sub-clusters.

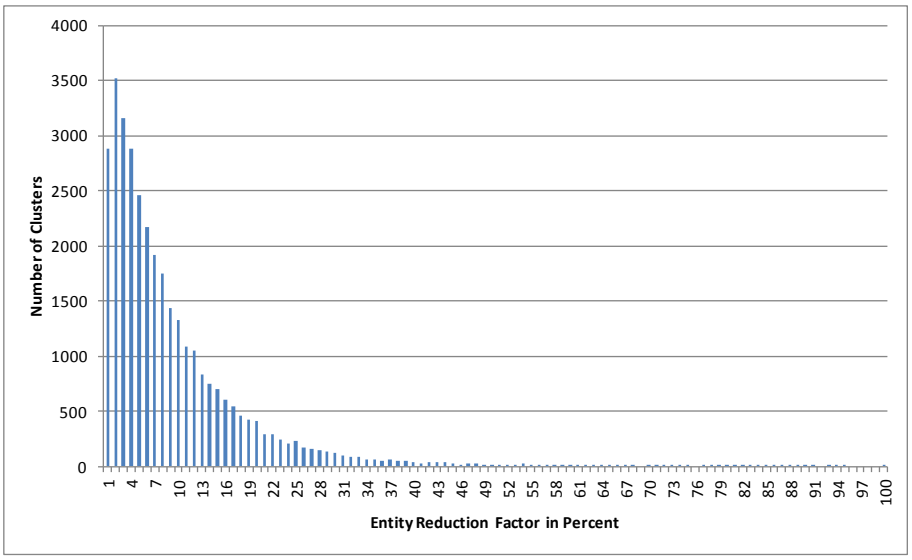


Fig. 26. Number of clusters including x percent of the entities for k=12 compared to k=1

Fig. 26 shows the number of clusters including a certain percentage of entities for k equals 12. For example, there are 3500 sub-clusters where the number of entities has been reduced to 3% of the number of entities for k equals 1. This observation is quite important considering a facetted-search scenario. For example, in our ViFaChem2 portal[21] the user has the possibility to decide for relevant chemical entities after submitting a query. Our evaluation has shown that this set of offered chemical entities can be highly decreased by only considering entities from the same sub-cluster leading to a more sophisticated search experience.

3.3. Conclusions

There are many different similarity measures available in chemistry. All rely on a fingerprint representation of the chemical structure. We evaluated the correlations between 16 widely used similarity measures and 6 different fingerprints for chemical entities using Kendall's Tau. The results show that many of them are uncorrelated, meaning they deliver different rankings.

Since chemistry is a wide field with many different sub-domains, these results seemed reasonable. Chemists are focused on specific tasks when searching for literature, for example drug design or synthesis. We have analyzed whether the uncorrelated measures fit to typical search tasks in chemistry. The different fingerprints

[21] www.chem.de

represent different chemical aspects. For example, the Substructure fingerprint only considers the structure of a molecule, whereas the MACCS fingerprint uses a set of questions regarding more properties of a molecule than just the structure. We investigated if it is possible to assign one similarity measure to one specific task. We conducted a user study with domain experts and have shown that for the same task, e.g., drug design, different domain experts preferred different similarity measures. Hence, it is not possible to assign one similarity measure to one specific task, meaning there is no similarity measure always delivering the most suitable result set for that task. During discussions with domain experts, we discovered that chemists usually have special background knowledge when searching for literature that cannot be expressed in the query.

One possible solution is to build a personalized retrieval system learning the most preferred measure for each individual chemist. We provided a system based on user feedback. Our evaluations showed that it is indeed possible to learn the most preferred measure within a couple of feedback cycles. Nevertheless, we also analyzed how to model the chemists' implicit knowledge to further improve the retrieval quality. We figured out that chemists are usually interested in chemical entities having a specific reaction behavior and belonging to a specific chemical class. The reaction behavior of a chemical entity can be determined by analyzing its structure and extracting its so-called functional groups, which are responsible for the entity's characteristic reaction behavior. Currently, only a few knowledge bases are available allowing for an automatic association of chemical entities to chemical classes. All are focused on small sub-domains of the whole domain of chemistry. A prime example is the ChEBI ontology, covering chemical entities and classes that are of biological interest. Thus, the chemical class is mainly based on the implicit knowledge of the chemist.

We presented an approach, clustering chemical entities based on their functional groups to externalize the chemists' implicit knowledge. The resulting clusters were manually analyzed by domain experts. The result was that the clusters including most of the entities are too unspecific and do not fit to the chemists' perception of chemical classes. Therefore, we used fingerprint-based similarity measures to further divide these clusters into sub-clusters. Since many uncorrelated combinations of fingerprints and similarity measures are available, we analyzed which one is the most promising. Our evaluation has shown that the Substructure fingerprint in combination with the Manhattan distance retrieves the best results for a query entity, ranking almost all top-ranked entities in the same functional groups cluster. We used this combination for building sub-clusters. Since we used k-means clustering, we also evaluated which value for k should be used for our collection. As ground truth, the ChEBI ontology was used associating chemical classes to each cluster. The optimal decomposition is found if each cluster has exactly one chemical class assigned. Whereas ChEBI only covers a small part of all chemical classes, we were able to enhance the number of classes by an order of magnitude. Even though, we cannot

assign explicit class names to the resulting clusters, they reflect the chemists' perception of chemical classes. We also evaluated the functional groups cluster in a retrieval scenario. By considering the implicit knowledge of each chemist, the result set can be further decreased without influencing the retrieval quality. Almost all relevant chemical entities are located in the same sub-cluster as the query entity. This leads to a decrease of the entity set of around 90% in average without losing relevant entities. This is quite useful for, e.g., faceted browsing.

Chapter 4

Contextual Search

In the last chapter, we analyzed different similarity measures and tried to assign them to different search tasks. We defined the search task as the working field of the respective user, like, e.g., drug design. However, it was not possible to assign the similarity measures to search tasks, since the users have some kind of implicit background knowledge influencing their subjective notion of relevance. We modelled this knowledge by clustering chemical entities based on their functional groups.

In this chapter, we further specify the idea of the search task. In entity centric searches, users are often interested in information about the query entity in a certain context. We define a context as any general concept that is semantically related to the query entity. For example, consider a user who is interested in chemical entity *Sildenafil* in the specific context of *irregular heartbeat*. This context is important to increase the retrieval precision. Certainly, it is possible to build a search engine, which has the ability to combine the entity and context as a query term. A simple workflow dealing with these combined queries is shown in **Fig. 27**.

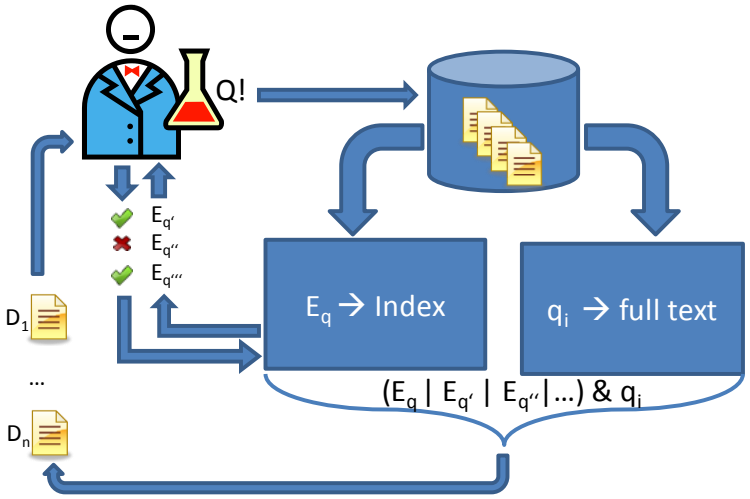


Fig. 27. Advanced workflow considering context

Here, the combined user query *Q!* is sent to the search engine and split up into the chemical entity E_q and the specified context q_i. We already solved the problem of finding similar entities regarding E_q. Here, we want to restrict the result set to the desired search context. One easy solution would be to filter the documents using an inverted fulltext index and only use those documents including q_i. However, as we will see in our evaluation, such a simple filtering also with expansion terms is not sufficient.

The general problem of contextual searches is that users usually describe their broad information needs with several keywords, which are likely to be different from the words used in the actually relevant documents. As a consequence the results returned by the information provider may miss relevant documents with respect to the user's information needs. This leads to a dramatically decreased retrieval quality and thus a bad usage experience. To guarantee high quality retrieval it is therefore important to bridge the gap between the query terms and the documents' vocabulary. The challenge of expressing the user's information need by finding the right query terms is widely known as the vocabulary problem [47]. Users often try to solve this problem by refining their query, i.e. adding or changing query terms in case the retrieval results have not been satisfying [48]. However, considering scenarios where users are searching for information about abstract concepts the problem of word mismatch is even bigger: such abstract concepts or context terms hardly ever occur directly in Web documents. Imagine a user who is interested in *information retrieval*. By entering the conceptual query '*information retrieval*', he only receives documents dealing with this very general concept. Closely related and also relevant documents not containing the exact term, like, for instance, documents about *Web search*, will not be returned. This also holds for more specific conceptual queries, like, e.g., *polyomavirus infections* in the biomedical domain or searches for chemical classes, like, e.g., *alcohol*, in the domain of chemistry.

To solve this problem, in some domains documents are already pre-annotated with suitable context terms. The most prominent example is the MEDLINE corpus, which is currently the largest document repository of life science and biomedical documents, containing more than 20 million publications. Each of these documents is manually annotated with several terms from the Medical Subject Heading (MeSH) ontology, which offers a controlled vocabulary for indexing and retrieval purposes. However, document collections like MEDLINE are a rare case and most collections lack suitable context annotations. For most domain specific collections no suitable controlled vocabularies or even better, ontologies, are available.

In this chapter, we use external knowledge provided by Wikipedia to semantically enrich documents bridging the gap between contextual queries and documents' vocabulary. We extract the most important terms from each document and enrich them with several semantic features gathered from Wikipedia. The enriched terms are used to compute the relevance of a document to a contextual query. Our experiments show that our approach outperforms traditional query expansion methods using statistical query expansion, showing an increase of more than 30% in Mean Average Precision. We also compare against stronger baselines using LSA and random indexing showing an improvement of more than 15%. All results have been proven to be statistically significant.

In chemistry, another problem is that none of the structure-based similarity measures takes such context information into account. But, also here this is very

important, because the similarity of two chemical substances is actually heavily related to the search context. Considering, for instance, the chemical entities *Zanamivir* and *Ibuprofen*, both are used in the treatment of flu and are therefore similar regarding this pharmacological activity context. *Ibuprofen* is also used to treat inflammatory diseases such as rheumatoid arthritis. But, regarding this context both entities are very dissimilar: *Zanamivir* is a neuraminidase inhibitor and thus not at all useful for the treatment of rheumatoid arthritis. It is therefore necessary to personalize measures for entity similarity to the search context a user is currently engaged in. In brief, context used to disambiguate the user's explicit query can be expected to lead to focused and relevant retrieval results.

The traditional way of searching for documents relevant for contextual queries is to use query expansion. It expands the query term issued by a user with suitable related terms, called expansion terms, matching the documents' vocabulary. In general, query expansion leads to higher recall, but strongly decreases the retrieval precision. The reason is that usually the context of the query is not known and thus the expansion terms do not meet the user's search intention. More advanced retrieval models, like Latent Semantic Analysis (LSA), try to solve this by producing sets of concepts related to the documents and their contained terms. However, as we will see in our experiments, the resulting quality is still not sufficient. For digital library providers it is important to enable contextual queries while also ensuring their high quality requirements.

One famous ranking algorithm considering context information for Web searches is the topic-sensitive PageRank [49]. For each Webpage multiple importance scores with respect to various topics are computed. These scores are combined at query time dependent on the topics stated in the query. Afterwards they can be combined with different IR measures to produce a suitable ranking.

Use Case: Imagine a chemist from the field of infection research who is currently working on a chemical substance called *Clindamycin*. He is interested in literature in the context of Malaria to find similar substances to overcome side effects, like, e.g., diarrhea or nausea.

Since *Clindamycin* is a lincosamide antibiotic it is usually used to treat infections with anaerobic bacteria. But, it can also be used to treat some protozoal diseases, like, for example, Malaria. Our chemist is interested in similar substances to *Clindamycin* that are used in the context Malaria. Therefore, he is searching for documents dealing with Malaria that may also include the most similar substances. One possible way for including the search context is to filter the document set and only consider documents containing the respective context term. The similar entities are found using fingerprint-based similarity measures.

The question is if we found the most similar chemical entities regarding the search context using fingerprint-based similarity measures? The problem is that context similarity cannot be captured in similarity measures based on structural

information. Two chemical entities can be very similar regarding one context, but totally dissimilar in other contexts.

For example, let us consider the chemical substance our chemist is searching for, *Clindamycin*. He is interested in similar entities in the context of Malaria. One very good hit in this context is *Quinine*. But, if we do not consider the context, and only use structural information both entities are very dissimilar. **Fig. 28** shows the chemical structures of both entities.

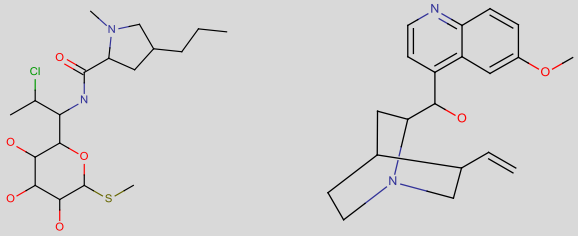


Fig. 28. Chemical structure of *Clindamycin* (left) and *Quinine* (right)

We searched for similar entities for *Clindamycin* in the PubChem database[22] using a structure similarity search and analyzed the results. The chemical entity *Quinine* was not under the top 1000 most similar substances.

To proof this assumption, we did an experiment on the PubMed central document corpus (PMC) containing 213,516 documents. Each document in that corpus is annotated with MeSH terms, which we used as context terms for this experiment. We choose ten query entities, each of them occurring in two different contexts. For example, we searched for *Clindamycin* in the contexts of *Malaria* and *Pregnancy* or *Gatifloxacin* in the contexts of *Aspergillose* and *Tuberculosis*. We took all documents for each query entity and context term and extracted the included chemical entities using OSCAR. Furthermore, we did a similarity search without respecting the context in PubChem retrieving all chemical entities with a similarity value higher than 80% to the respective query entity. Since a chemical structure can have several names, we also included all synonyms in the PubChem set. We compared the different sets using Jaccard similarity. We observed that the PubChem set and the respective context sets have only a very few entities in common, resulting in an average Jaccard similarity of only 0.0008. We also computed the similarity between the two context sets for each query. Of course, the similarity value depends on how the two context terms are semantically related. The closer they are, the more similar are the entity sets. But, also here the average Jaccard similarity is low with only 0.19. We discussed our observations with domain experts, which confirmed that structural similarity is usually not the determining factor for context searches. Nevertheless, it is used for similarity searching and

[22] http://pubchem.ncbi.nlm.nih.gov

the retrieved result set is filtered using facetted browsing to focus on the search context.

This experiment leads to two observations:

- Structure-based similarity measures are not the right choice for performing context searches in chemistry.
- Since the result sets for the same query entity in different contexts strongly differ, the context terms are important for finding similar entities.

In this chapter, we present two approaches allowing for contextual searches in chemistry. The first uses knowledge harvested from Wikipedia as a major knowledge base. One advantage of that approach is that every term occurring in Wikipedia can be used as context term. Instead of using a fixed vocabulary of predefined classes, we thus use the 'wisdom of the crowd', which is dynamic and ever growing. The derived similarity measure is therefore not purely based on structural information of chemical entities, but extracts different features of chemical entities using common knowledge in the community. All features are combined in enriched profiles of chemical entities. These profiles are then used for similarity computations resulting in a personalized ranking function considering both, context as well as entity similarity.

The second approach uses cross-domain knowledge from different, but related domains, to annotate documents with suitable context terms. For chemistry, we take documents from the biomedical domain which are annotated with MeSH terms, learn suitable classifications and annotate chemical documents.

4.1. Related Work

In general, the area of automated text categorization is a wide field dating back to the early '60s. Central approaches in the '80s were usually based on knowledge engineering, where a human expert defined a set of rules to classify documents under the given categories. Due to the machine-learning paradigm and more powerful hardware devices the knowledge engineering approach lost popularity in the research community in the '90s. In machine-learning a general inductive process automatically builds a classifier by learning the interesting characteristics from a set of pre-classified documents. Nowadays, text categorization plays a major role in information systems and is applied in many contexts, like, e.g., document indexing or filtering, automated metadata generation or word sense disambiguation. An overview of machine learning in text categorization is given in [50].

In reality, almost all queries are either implicitly or explicitly formulated in a specific search context. For an implicit context, factors, like, e.g., the user's domain of interest, knowledge or preferences, are important to get the correct interpretation of the query. For explicitly formulated contexts in the query it is important to solve problems, like, for example, the vocabulary problem [47]. In many studies it was

proven that context terms are important for high quality retrieval, because they have a strong influence on the users' relevance judgments, see [51], [52], [53], [54], [55].

4.1.1. Extending the Query with Implicit Context Information

Today, there are several groups of approaches using implicit context information. One of the newest groups of approaches tries to automatically detect the current situation a user is in. In [56] the authors introduce an approach using sensors automatically detecting what a user is currently doing. They argue that the search results should differ dependent on the current user activity. The authors in [57] proposed a method automatically searching for a Webpage related to the daily activity of a user. They construct a query considering the use of daily objects employed in the activity that is detected with object-attached sensors.

Actually, there are many overlapping areas trying to extend the user's query with context information, like, for instance, query expansion, conceptual retrieval, or cluster-based retrieval. The aim of query expansion is to bridge the gap between queries and documents by adding additional terms or reweighting terms in the original query [58]. In general, query expansion approaches can be local or global [59]. Local methods try to consider, e.g., the user's search history or profile, to automatically enrich queries [60]. These systems do not consider the physical surrounding of the user, but also try to infer context terms automatically. Also relevance feedback is a form of local query expansion. Here, the retrieved documents are examples to find additional query terms [61]. In the area of pseudo relevance feedback, it is assumed that the top-k retrieved documents are relevant. But, it is also possible to consider implicit or explicit relevance judgments from users, see, e.g., [62] or [63]. An approach using information from raw query search logs to discover context terms is described in [64]. The detected terms are included in user preferences used to optimize search results. It was shown that in terms of top-k search quality a system using context information outperforms existing personalization approaches without context information. In [65] a model based on language modeling is presented, where the context specification is done based on the query and not on the user. If users are interested in different domains, using, for example, user profiles can sometimes lead to a query drift favoring incorrect documents. This can be avoided using query-specific contexts.

Global expansion approaches use global collection statistics or external knowledge sources, such as concept languages, to enhance the query. There are many approaches using concept relations defined in a thesaurus. In [66] term relationships are used to extend the query model. In contrast to other language models they do not assume term independence. While considering relationships between terms, e.g. synonymy, the retrieval performance is enhanced. A combination of local and global approaches is presented in [67]. Here, a local expansion method is used to obtain a conceptual representation of a query. Afterwards, a global method is used to translate the conceptual representation back to textual representation. To

get the conceptual representation of the query pseudo relevance feedback is used and the query is translated into the set of concepts associated to the relevant documents. Using these concepts the user's information need is represented on a higher, conceptual level. To get better retrieval performance this conceptual query model is translated back to a textual model.

In [68] three different algorithms are compared considering contextual search for the Web, i.e. query rewriting, rank-biasing and iterative filtering meta-search (IFM). For query rewriting each query is enriched with appropriate terms from the search context and for answering this augmented query an off-the-shelf search engine is used. The initial query is expanded using all terms from a context term vector formulating a rather long query using AND semantics. The rank-biasing algorithm generates a representation of the context and answers queries using a custom-built search engine exploiting this representation. For the IFM algorithm multiple sub-queries based on the initial query and appropriate terms from the search context are generated. Each of these sub-queries is sent to a search engine and the results are re-ranked using rank aggregation methods. The advantage of the simple query reformulation algorithm is that it naturally fits with the search interfaces offered by major search engine providers and therefore can be implemented on top of them in a straight forward fashion. And also the experimental results in [68] have shown that this approach performs surprisingly well. Therefore, we will compare against a quite similar approach using query expansion for the context term in our evaluation.

4.1.2. Extending Documents with Context Information

In [69] it was shown that context-sensitive ranking improves the retrieval quality for domain experts remarkably, compared with conventional ranking models. The ranking model uses keyword statistics collected from the specified contexts to rank the documents. To reduce the problem of computing keyword statistics at runtime for the document subsets of the specified context the authors suggest using materialized views. It was shown that the materialized view technique improves the efficiency of worst case queries significantly. The technically difficulty is to choose a small number of materialized views to improve the system overall performance. A data mining based and a graph decomposition based algorithm have been presented to solve the selection problem. Since they are working on the MEDLINE corpus, all given documents are already pre-classified by the annotated MeSH terms. A more advanced approach described in [70] uses semantic information extracted from texts and some domain ontology to approximate concepts associated with documents. Since for document classification, respectively context annotation, it is necessary to know the set of possible classes in advance, using the controlled vocabulary and semantic relations of an ontology is beneficial.

Actually, in the biomedical domain almost all documents are annotated with one or more terms from the MeSH ontology. This ontology defines a controlled vocab-

ulary specifying a variety of concepts in (biomedical) science. Each MeSH term represents a concept and a combination of these terms represents the context spanning the corresponding concepts. A researcher can use tools that visualize the MeSH ontology for specifying his/her search context by browsing through the ontology and selecting terms that are relevant for his/her context. An example is the GoPubMed[23] portal where the user can do facetted searches by navigating through the MeSH ontology and filter the PubMed document corpus by choosing suitable ontology terms. In [71] it is also shown that the MeSH ontology is a valuable resource for representing MEDLINE documents at different abstraction levels. The authors evaluated the suitability of the ontology for classifying biomedical documents using a Bayesian Network classifier. Furthermore, it was shown that the classification accuracy can be improved by increasing the number of MeSH terms used for representing a document. Another approach trying to extend the ontology-based representation of biomedical documents is described in [72]. The initial MeSH annotations of biomedical documents have been extended with semantically similar concepts from the MeSH ontology. A simple edge-count similarity measure was used to evaluate the semantic proximity between different concepts.

In [73] an approach is presented focusing on the automatic annotation of MeSH terms to biomedical documents. Different classification systems are compared to reproduce manual MeSH annotations. Experiments also showed that the retrieval quality for biomedical documents can be improved by automatically annotating the user query with MeSH terms. A similar approach dealing with automatic query expansion in MEDLINE but using a pseudo-relevance feedback technique is described in [74].

But, the almost completely MeSH-indexed MEDLINE digital library is a rare case and its manual curation is expensive, while automatic classification is still error-prone. Moreover, most document collections miss both, suitable annotations and the funds to add them. Considering, for instance, the linked open data community, hardly any collection dealing with chemical entities is properly annotated. Examples are *Linking Open Drug Data*, a task force within the World Wide Web Consortium's Health Care and Life Sciences Interest Group, or *clinical trials* describing relationships between active ingredients and diseases tested in clinical studies around the world.

In the chemical domain, the most comprehensive database is still created manually by the Chemical Abstracts Services (CAS) as part of the American Chemical Society. Although some Web portals for searching for chemical documents are freely available, like, e.g., ChemXSeer[24] or the ViFaChem portal[25], none of them allows for con-

[23] http://www.gopubmed.com

[24] http://chemxseer.ist.psu.edu

[25] www.chem.de

text-aware retrieval. The reason is that there is no suitable knowledge base in chemistry offering a defined vocabulary comparable to the MeSH ontology in the biomedical domain. A possible approach might be the automatic creation of ontologies. But, unfortunately, the quality of automatically generated ontologies for such complex domains as chemistry is not yet sufficient [75].

Our idea is to use the knowledge from different, but related domains or the knowledge from general information sources, like Wikipedia, either to annotate documents with suitable context terms, or to compute context similarity on-the-fly.

The approach in [76] discusses the problems of cross-domain knowledge transfer. The main focus lies on the problem that for classification training and test data have to follow the same distribution. Since for cross-domain classification this is usually not the case a two-stage algorithm is presented based on semi-supervised classification. In [77] an approach enabling cross-domain search by exploiting Wikipedia is shown. The focus is on analyzing tags used in Web 2.0 systems like Flickr and connect them to concepts in Wikipedia. Other approaches use Wikipedia directly to improve document retrieval. In [78] an approach is presented using machine learning techniques with Wikipedia to enrich document retrieval. The same authors presented a concept-based retrieval approach based on Explicit Semantic Analysis (ESA) in [79]. Their results show the usefulness of Wikipedia to compute semantic relatedness of natural language text. Another approach presented in [80] uses Wikipedia concept and category information for enriched document clustering.

4.2. Combining Entity and Context Similarity Using External Knowledge Bases

One drawback of all fingerprint-based similarity measures is that they always need structural information about chemical entities, like, e.g., the SMILES representation. Moreover, for computing context dependent entity similarity structural information is not sufficient. Therefore, we introduce a similarity measure that uses external knowledge and is independent of the chemical structure. Our measure considers both, entity- as well as context similarity. Finally, we are interested in documents including the query entity (or similar entities) in the sense of the specified context.

In our system a document, further denoted by d, is represented as the bag of words of its included chemical entities $E_d \subseteq E$, where E is the set of all chemical entities in the collection. Let D denote the collection of documents. A query for a context search is composed of two parts: $q = e_q \mid q_c$, where e_q is a chemical entity and q_c is the desired context specification. q_c specifies a sub-collection $D_c \subseteq D$ such that $\forall d \in D_c$, d satisfies q_c. The basic workflow is shown in **Fig. 29**.

The automatic entity extraction is done using the OSCAR framework. The documents and their contained chemical entities are stored in a database. For each chemical entity, a chemical profile is created containing different features. The basic knowledge base used is Wikipedia. For each chemical entity $e \in E_d$, we analyzed its

corresponding Wikipedia page and extracted suitable features used in the chemical profiles. From each page, we extracted the set of the assigned Wikipedia categories. In addition, a set of all other entities that are cited in the Wikipedia page (outgoing links) and a set of all other entities pointing to the respective page (incoming links) are used in the profile.

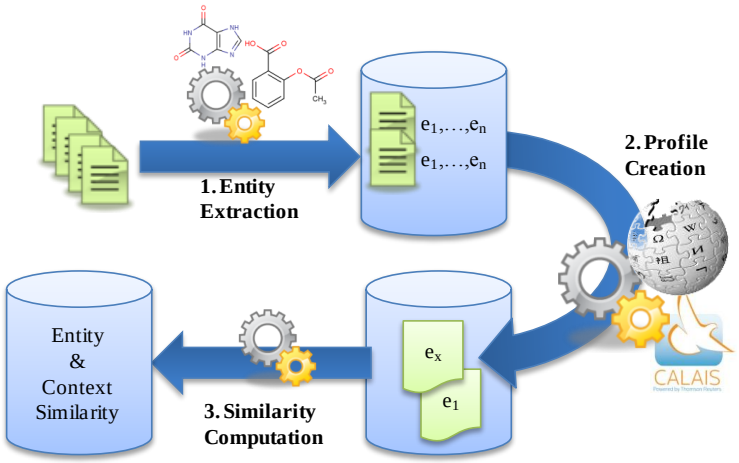


Fig. 29. Information extraction process

Beside Wikipedia, we also use another tool to automatically detect important entities in text, named OpenCalais[26]. OpenCalais is a free Web service from Thomson-Reuters that does named entity recognition to extract events and relationships from text. It uses natural language processing and machine learning techniques to recognize instances of named entities. Since OpenCalais uses surface features, like, e.g., capitalization, and is not based on handcrafted databases of entities, it can detect new entities that may not be included in any knowledge base like Wikipedia.

For each chemical entity, we analyze its Wikipedia page using OpenCalais and add the retrieved information to its chemical profile. In detail, we use the detected Calais entities, topics and social tags. The Calais entities are further divided into several different types, ranging from types like medical treatment or medical condition, to types like person or operating system. The social tags are not really semantic features, but emulate how a person would tag a specific piece of content. The topics describe a category that the input content is about. They are based on the Calais categorization taxonomy. But, it is also possible that no topic is assigned to the input content.

All extracted features are written to the chemical profile of the entity and stored in our system. In the next step, the similarity between all chemical profiles is precomputed using the similarity measures described in the next section. To fasten the retrieval process we also pre-compute some context similarities. As given context

[26] http://www.opencalais.com

terms we use the Wikipedia categories associated with the entities' pages. For each distinct extracted Wikipedia category (context term) the context similarity to all chemical entities is computed and stored in a database. Please note that this step is not necessary, but done to save computation time during query processing. One advantage of our system is that the user is not bound to the fixed vocabulary of some context ontology used for classifying the documents into search contexts. With Wikipedia, we use the wisdom of the crowd that is growing fast. If the user enters a context term that is not known in our system the context similarity is computed and stored at query time. However, only terms having their own Wikipedia page can be used as context terms.

4.2.1. Entity Similarity

Each chemical profile contains six different features. Each feature is used to compute the similarity between the query entity e_q and the entity $e_a \in E$.

Calais entity similarity: Let ts_q be the type set for e_q and ts_a the type set for e_a. Each type $t \in ts_x$ where $x \in \{q, a\}$ is associated with a set of related Calais entities, $t_n es_q$ and $t_n es_a$, where $1 \leq n \leq |ts_x|$. The similarity is computed using the Jaccard coefficient.

$$ts = \frac{ts_q \cap ts_a}{ts_q \cup ts_a} \tag{2}$$

The *ts* coefficient describes how many types the given chemical entities have in common. For each type they have in common the entity similarity is computed and normalized by the number of types e_q and e_a have in common.

$$es = \frac{\sum_{t \in ts_q \cap ts_a} \frac{t_t es_q \cap t_t es_a}{t_t es_q \cup t_t es_a}}{|ts_q \cap ts_a|} \tag{3}$$

The Calais entity similarity is computed as follows:

$$ces = (\gamma * ts) + ((1 - \gamma) * es) \tag{4}$$

where γ is a weighting factor and $0 \leq \gamma \leq 1$.

Calais tag and topic similarity: For tag and topic similarity, the same measure is used. For each detected term (tag or topic term) a relevance score in the range of 0 to 1, further denoted as *rs*, is computed, describing the importance of each unique term.

Let tsm_q be the term set for e_q, and tsm_a the term set for e_a. The tag and topic similarity is computed using the following equation:

$$tsm = \beta * \frac{tsm_q \cap tsm_a}{tsm_q \cup tsm_a} \tag{5}$$

β is called the regulation factor which is computed as follows:

$$\beta = \frac{\sum_{t \in tsm_q \cap tsm_a} \frac{rsa_t + rsq_t}{2}}{|ts_q \cap ts_a|} \quad (6)$$

where rsa_t is the relevance score of term t for e_a and rsq_t the relevance score of t for e_q. The relevance scores are in the range of 0 to 1 and are assigned by OpenCalais. The regulation factor is used to give lower similarity scores to entities that indeed have many terms in common, but which have low relevance scores for the entity itself.

Wikipedia category similarity: For the Wikipedia category similarity, we defined a quite similar formula as for the Calais tag and topic similarity. Let wc_q be the categories set for e_q and wc_a the categories set for e_a. For each Wikipedia category, also a weighting factor (*wf*) is assigned describing how general the respective category is regarding the Wikipedia category graph. We use this factor to give more specific categories a higher score. The category similarity is computed using the following formula:

$$wc = wf * \frac{wc_q \cap wc_a}{wc_q \cup wc_a} \quad (7)$$

The weighting factor *wf* is defined as

$$wf = \frac{\sum_{wc \in wc_q \cap wc_a} dt_{wc}}{|wc_q \cap wc_a|} \quad (8)$$

where *dt* is the length of the shortest path from the respective Wikipedia category to the root category.

Wikipedia related entities similarity: Furthermore, we use the Jaccard coefficient to compute the similarity based on the related entities. For related entities, we distinguish between entities linking to the Wikipedia page of e_a and e_q (further denoted as res_{in}) and entities that are linked from the Wikipedia pages of e_a and e_q (further denoted as res_{out}).

Let res_q be the set of related entities for e_q and res_a the set of related entities for e_a. The similarity is computed as follows:

$$res_{in/out} = \frac{res_q \cap res_a}{res_q \cup res_a} \quad (9)$$

Entity similarity: To compute the entity similarity of e_a and e_q we combine the different feature similarities in a linear fashion.

$$entSim = \omega * ces + \vartheta * tsm_{tag} + \sigma * tsm_{topic} + \vartheta * wc + \rho * res_{in} + \tau * res_{out} \quad (10)$$

Each feature is multiplied with a Boolean variable, i.e. $\omega, \vartheta, \sigma, \vartheta, \rho, \tau$, having the value 0 or 1. These variables are used for personalizing the entity similarity measure by switching features on and off. As we will see in the retrieval experiments, it depends on the user preferences which combination of features leads to best retrieval results.

4.2.2. Context Similarity

The context similarity is also based on the knowledge covered by Wikipedia. We use the Wikipedia Miner [81] to access the Wikipedia corpus and compute the semantic similarity between the context term and all chemical entities in our corpus using the relatedness measure described in [82]:

$$contextSim(c, e) = \frac{\log(\max(|C|,|E|)) - \log(|C \cap E|)}{\log(|W|) - \log(\min(|C|,|E|))} \qquad (11)$$

where *c* and e are the Wikipedia pages for the context term *c* and the entity e, *C* and *E* are the sets of pages that link to *c*, respectively e, and *W* is the set of all pages in Wikipedia.

A drawback of this measure is that we need to compute the semantic similarity between the context term and all other chemical entities in our collection. After computation the scores are stored in a database meaning that we only need to compute the similarity once for every context term. In case a new context term is entered in the system, this computation has to be performed. The next time the context term is entered no computation is necessary and the scores can be directly retrieved from the database.

4.2.3. Combined Similarity

Our goal is to find the most similar entities for the query entity e_q in the given context q_c. The entity similarity computes the most similar entities for e_q and the context similarity finds the most related entities to the context term. The total similarity for query *q* is computed as follows:

$$totalSim = (\alpha * contextSim) + (1 - \alpha) * \frac{entSim}{|EF|} \qquad (12)$$

where *EF* is the set of features used for entity similarity computation and α is a weighting factor with $0 \leq \alpha \leq 1$.

4.2.4. Evaluation

For our experiments, we used a data set of 44660 clinical studies[27]. The dataset includes documents ranging from the year 1908 to 2015. The documents from 2015 are planed clinical studies that have not yet started. We choose 10 different context terms, which are all diseases, i.e. Malaria, Tuberculosis, Mumps, Tinnitus, Hypertension, Hepatitis A and C, Influenza, Dengue and Cancer. We automatically extracted all chemical entities using the OSCAR framework. In total 1.573.264 entities have been annotated in the documents, 79223 of them are distinct. Since we want to compare against the fingerprint-based similarity measures we filtered out all found entities that do not have structural information (in this case a SMILES code). This

[27] http://clinicaltrials.gov/ct2/home

leads to a total of 721 distinct chemical entities independent of the documents' context.

Within our experiments, we evaluate the following statements:

- First, we analyze if it is sensible to combine all features in a ranking function or if any of them are correlated. Therefore, we created an experiment using the KTau correlation coefficient to compare the different feature rankings.
- Since our experiment in the use case has shown that structure-based rankings are no suitable option for chemical context searches, we compare the feature rankings to all uncorrelated fingerprint-based similarity measures.
- Next, we compared the uncorrelated fingerprint-based similarity measures to the feature-based similarity measure computing Mean Average Precision. Thus, we created an experiment where a group of domain experts manually assessed the relevance of chemical entities regarding a query consisting of chemical entity and context term.

Correlation of Features

Since our similarity measure is based on Wikipedia knowledge, we first analyzed how many of the chemical entities can be found in Wikipedia. We used the WikipediaMiner to search for the chemical entities in Wikipedia. For 92.6% (668) we found a matching Wikipedia page. Furthermore, we analyzed if we need all features in the chemical profile for similarity computation or if some of them are correlated. We randomly chose around 10% of all chemical entities as query terms, resulting in 72 queries in total. Using these terms we computed the rankings to all other chemical entities in our set based on the six feature similarities.

Since we can interpret the similarity value as a value in a ranking vector, we used the Kendall rank correlation coefficient (KTau) to determine the correlation of the different measures. We calculated the correlation coefficient for each ranking vector and the arithmetic mean over 72 queries. A KTau of 1 means that the agreement of two rankings is perfect, -1 indicates a perfect disagreement and for independent rankings one would expect the coefficient to be *approximately* 0. For each pairwise comparison of two rankings we averaged the Ktau values over all queries. We only considered those queries which are significant meaning having a p-Value less than 0.05. **Table 9** shows the KTau values for each pairwise comparison of all features.

Table 9. KTau values for features

	ces	**tsm_{tag}**	**tsm_{topic}**	**wc**	**res_{in}**	**res_{out}**
ces	1	0.04	0.02	0.04	0.04	0.05
tsm_{tag}	0.04	1	0.08	0.13	0.10	0.11
tsm_{topic}	0.02	0.08	1	0.41	0.14	0.08
wc	0.04	0.13	0.41	1	0.44	0.36

	ces	tsm_{tag}	tsm_{topic}	wc	res_{in}	res_{out}
res_{in}	0.04	0.10	0.14	0.44	1	0.34
res_{out}	0.05	0.11	0.08	0.36	0.34	1

The highest correlation is found between the Wikipedia in-links and the Wikipedia categories, followed by the Open Calais topic ranking and the Wikipedia categories. However, the values are still very small so that we consider the rankings as uncorrelated. Therefore, all features deliver different rankings and are used in our similarity measure.

Correlation of Feature-based and Fingerprint-based Measures

In a second experiment, we evaluated if any of the feature rankings correlate with any of the uncorrelated fingerprint-based similarity measures. We used the 72 queries and the corpus of 721 chemical entities and compute the different rankings. Again, we compare the rankings using the Kendall Tau correlation coefficient. **Table 10** shows the KTau values for all similarity measures based on the substructure fingerprint.

Table 10. KTau values for similarity measures for substructure fingerprint compared to features

FP/SM	wc	res_{out}	res_{in}	ces	tsm_{tag}	tsm_{topic}
Forbes/ Substructure	0.03	0.00	0.05	0.02	-0.01	0.10
Manhattan/ Substructure	0.06	0.04	0.02	0.01	0.02	0.03
Russell Rao/ Substructure	0.11	0.03	0.03	-0.03	-0.01	0.11
Simpson/ Substructure	0.16	0.06	0.10	-0.01	0.07	0.07
Yule/ Substructure	0.04	0.08	0.06	0.05	-0.06	0.02

The KTau values are all around zero. That means that there is no correlation between the fingerprint-based measures to any of the profile features. We only show the substructure fingerprint as an example, but the results are almost the same for all other fingerprints. Next, we compare the combined entity similarity to all fingerprint-based similarity measures. **Table 11** shows the results for the KTau comparisons.

The KTau values are all around zero and there is no correlation between the different measures. Since we know that the fingerprint-based measures do not deliver suitable rankings for context searches, it is good to see that our measure does not correlate to any of them. Next, we have to prove that the produced rankings are sensible in a retrieval scenario.

Table 11. KTau values comparing fingerprint-based rankings and feature-based rankings: Substructure (1), Estate (2), Graph-Only (3), MACCS (4), General (5), Extended (6)

	1	2	3	4	5	6
Forbes	0.08	0.09	-0.09	-0.03	-0.10	-0.09
Manhattan	-0.12	-0.11	0.02	-0.11	0.09	0.07
Russel-Rao	0.21	0.20	0.18	0.20	0.17	0.18
Simpson	0.17	0.12	0.03	0.13	-0.02	0.06
Yule	0.17	0.14	0.05	0.14	-0.02	0.07

Comparing Different Rankings

In this experiment, we compare the rankings of the different similarity measures. As stated earlier, a query is defined as follows: A query for a context search is composed of two parts: $q = e_q \mid q_c$, where e_q is a chemical entity and q_c is the desired context specification.

Basically, we compared the feature-based similarity against the fingerprint-based similarity measures. Since the relevance ratings for two entities differ between different context terms, it is not sensible to evaluate the entity ranking without considering the search context. For considering the context in the fingerprint-based measures, we used the following procedure. The documents in our collection, further denoted by D, are filtered and only those related to q_c are retrieved. From this document set, denoted by D_c, the chemical entities are extracted and ranked using the different similarity measures. We evaluated different possibilities for building D_c. First, we use a Boolean approach where D_c contains all documents including the context term q_c. Second, we use an approach using statistical query expansion, where q_c is expanded using the most co-occurring terms.

For building a ground truth to compare the different rankings against, we randomly choose a set of 10 chemical entities and related context terms as queries. In order to make manual relevance assessment feasible, we pooled together the top-20 entities retrieved for each query and similarity metric. The relevance assessment was done manually by a group of domain experts. The experts marked for each query all chemical entities from the sampling sets that are relevant for the query in a Boolean fashion. To evaluate the rankings, we computed the Mean Average Precision (MAP) based on the relevance assessments.

First, we analyze the results of the Boolean retrieval model. The document set is filtered using q_c, meaning only documents are included containing q_c in the fulltext. The filtering was done using a Lucene fulltext index. **Table 12** shows the MAP values for the Boolean approach (SM means similarity measure and FP fingerprint).

Table 12. MAP values for fingerprint-based measures for the Boolean approach: Substructure (1), Estate (2), Graph-Only (3), MACCS (4), General (5), Extended (6)

SM | FP	1	2	3	4	5	6
Forbes	**0,31**	0,15	0,07	0,20	0,08	0,11
Manhattan	0,07	0,12	0,13	0,06	0,14	0,15
Russel-Rao	0,25	0,26	0,22	0,24	0,28	0,27
Simpson	0,29	0,17	0,06	0,21	0,09	0,13
Yule	0,29	0,27	0,25	0,25	0,27	0,26

The highest MAP of 31% is reached using the Forbes similarity measure based on the Substructure fingerprint. Since the entity set is filtered in advance using the Lucene context filter it is possible that we filtered out relevant entities. Therefore, we also computed the recall. The average recall using the Boolean approach is 82.7%. That means, indeed some relevant entities are filtered out. The reason is that not all relevant documents contain the context term in the fulltext.

For the second baseline approach, we use a retrieval model including statistical query expansion. We computed a term-to-term co-occurrence matrix based on our document set. We also considered the position of the term in the document, meaning two terms that are close together will get a higher score. Furthermore, we only use terms as context terms fulfilling a certain popularity threshold. Finally, the context term q_c is expanded with the top-10 co-occurring terms using the following retrieval model: Let $C=\{q_c,c_1,\ldots,c_n\}$ be the set including q_c and all expanded terms. The expanded context query is formulated as q_c *OR* c_1 *OR* … *OR* c_n, meaning all documents are returned containing q_c or any of the expanded terms. **Table 13** shows the results for the MAP computations.

Table 13. MAP values for fingerprint-based measures for the co-occurrence approach: Substructure (1), Estate (2), Graph-Only (3), MACCS (4), General (5), Extended (6)

	1	2	3	4	5	6
Forbes	0,17	0,07	0,04	0,17	0,05	0,09
Manhattan	0,02	0,04	0,01	0,01	0,03	0,04
Russel-Rao	0,18	0,14	0,15	0,17	0,21	0,20

	1	2	3	4	5	6
Simpson	0,14	0,05	0,04	0,14	0,06	0,07
Yule	**0,23**	0,22	0,21	0,22	0,22	**0,23**

It is interesting to see that the MAP is even lower than for the Boolean approach. The reason is that using query expansion the set of entities is getting bigger. This is also proved if we take a look at the recall. It has increased to 89.5%. These results indeed prove our assumption that fingerprint-based measures are not the right choice for context searches. In addition, they are also in accordance with the experiment shown in the use case scenario.

For the feature-based approach, we combined the context- and entity similarity in one single measure. Therefore, no filtering of documents is necessary. The similarities are computed for all chemical entities leading to a recall of 100%. To regulate the weighting between context- and entity similarity a variable *alpha* is used. If alpha is zero no context similarity is used and if it is one no entity similarity is used. **Fig. 30** shows the MAP results for the feature-based similarity measure for varying alpha values.

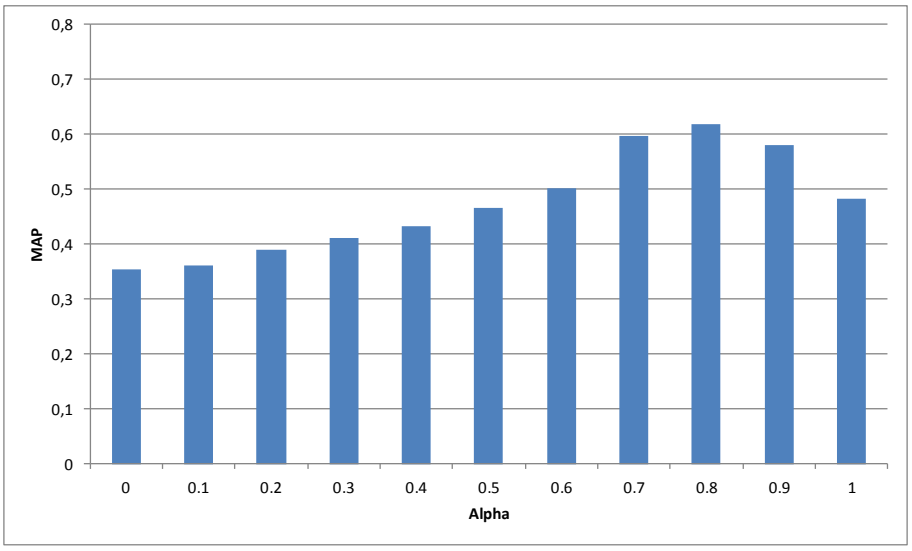


Fig. 30. MAP values dependent on alpha

The best result of a MAP of 61% is reached for alpha equals 0.8. That means the context similarity is slightly higher weighted. Using this measure, we were able to increase the MAP from 31% for the Boolean approach to 61%.

4.2.5. Retrieval Based on Feature-Based Context Similarity

Also for contextual searches, a personalized retrieval system improves the general retrieval quality. Each individual user trains the system and the system will learn the best similarity measure for the user. The system includes a simple feedback step

where the user marks the chemical entities most relevant for him. Therefore, we conducted a user study with domain experts from the area of drug design and synthesis. For the user study, we have randomly chosen ten queries consisting of chemical entity and context. Each query represents a feedback cycle in the system.

Since the measure for computing the entity similarity is composed of six different features, we analyzed which feature combination is the best for the individual chemist. The goal is to find a suitable feature combination for computing the entity similarity within the feedback cycles. Thus, we need to compute all possible combinations and analyze which leads to the best results. Let us consider we have a finite set *EF* containing *n* features. The number of different subsets we need to combine is computed using the power set, $|P(EF)| = 2^n$. Since we have 6 different features we can combine them in $2^6 - 1 = 63$ different ways. We need to subtract 1 since we do not need to compute the empty set, which is also contained in the power set.

For each chemist and each query, we computed the 63 different rankings and compared them to the manual relevance judgments by computing the average precision. For each query, we analyzed which feature combinations lead to the best result. Unfortunately, it was not possible to find the optimal solution for each chemist. But, we found out that in average four different feature combinations are enough to always find the most suitable ranking. These combinations have been found after seven feedback cycles in average. That means that we only need to compute four different rankings instead of 63 and have a high probability that the most suitable solution is found. **Fig. 31** shows the number of top rankings for the different feature combinations over all chemists. It is interesting to see that more than half of the combinations never lead to the best ranking.

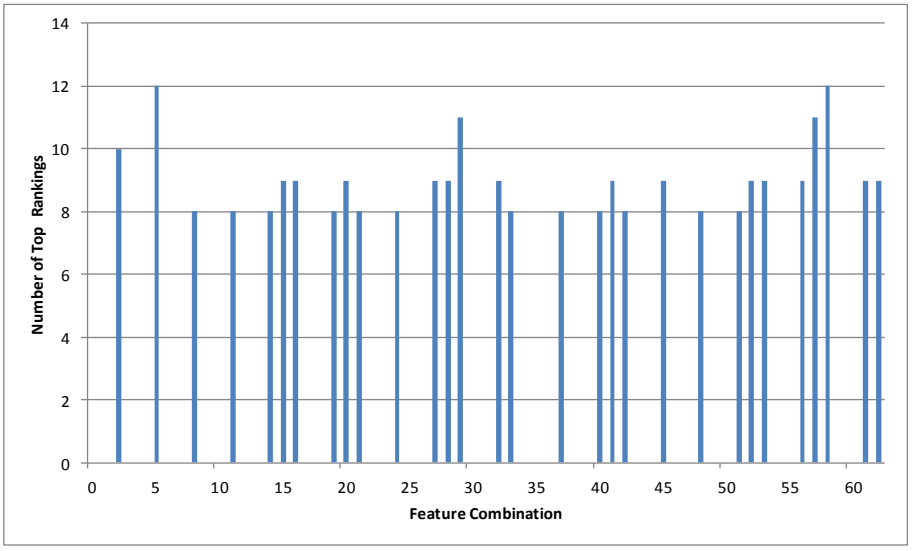


Fig. 31. Number of top rankings for different feature combinations

Of course, this statistic will change over time depending on the different users submitting queries to the system, but it is useful to overcome the well-known *new*

user problem. For new users coming to the system it seems to be a good choice to use the overall best measure as global starting point, i.e. tsm_{tag} and tsm_{topic} or res_{in} (see section 4.2.1).

Now that we found the best feature combinations, we use them to analyze which weighting between entity- and context similarity is the best by varying the alpha value. For each chemist and each query, we took the best feature combination and compute the average precision using the chemist's relevance vector. **Fig. 32** shows the MAP results for one chemist for varying alpha over 10 queries.

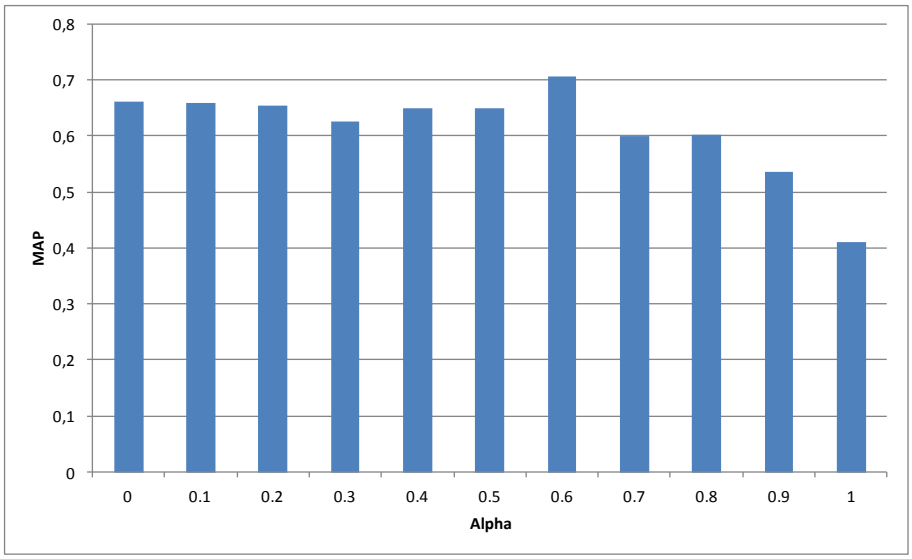


Fig. 32. Example: MAP values for varying alpha for one chemist over 10 queries

For this chemist, the best results are retrieved using an alpha of 0.6. Compared to the impersonalized measure the Mean Average Precision is increased of up to 71%. In average over all users, the Mean Average Precision increases about 9% using personalization.

4.3. Enriching Documents with Cross-Domain Knowledge

In the last sub-chapter, we presented a similarity measure combining entity- and context similarity. In this section, we focus on using knowledge bases with a fixed terminology for describing the search context. In general, information contained in terminologies (or more general: ontologies) forms very useful background knowledge for classifying documents in a context-aware fashion. For instance, in the *biomedical domain* the National Library of Medicine (NLM) uses the MeSH (Medical Subject Headings) ontology to annotate and index documents from biomedical journals [83]. MeSH defines a set of controlled vocabulary thesaurus including a set of description terms that are hierarchically organized. All these annotated documents are included in MEDLINE, which is currently the largest biomedical literature database. Web based interfaces have been developed to search over MEDLINE and

other related collections. The most commonly used Web interface is PubMed[28] comprising more than 21 million items of biomedical literature. However, MEDLINE indexed by MeSH is a rare case and is actually curated manually with expensive efforts. Most domains miss such overarching ontologies to annotate documents with suitable context information.

We show how to overcome the lack of context annotations for domains not offering general ontologies. The idea is to use cross-domain knowledge from different, but related domains. We extract named entities from documents annotated with ontology terms and train classifiers to predict these ontology terms based on the extracted named entities. Documents from related domains are annotated with ontology terms based on these classification models. To ensure that the annotated terms are semantically related to the documents' context a semantic processor is introduced. The semantic processor computes the semantic similarity between the associated ontology terms and the document's named entities to filter unrelated terms. This computation is based on a general knowledge base that acts as some kind of "glue" between the ontology terms from the source domain and the named entities used in the target domain.

We define the search context as any set of terms from the source ontology. If, e.g., a user is interested in documents relevant for a named entity in the context 'computer science', all sub-terms of the node 'computer science' from the source ontology are relevant. Thus, the search context can be very general, like 'computer science', but also very specific. It is only necessary to map this context to a set of ontology nodes.

4.3.1. MeTaSem – An Approach to Annotate Documents with Cross-Domain Knowledge

Our system consists of three main parts. An overview of our proposed workflow is shown in **Fig. 33**.

Model Extractor: First of all it is necessary to train classifiers to learn suitable models. Therefore, we take domain specific documents that have already been annotated with ontology terms and extract named entities. For example, we took MeSH annotated MEDLINE documents and extracted all chemical entities using the OSCAR framework. Afterwards, for each document, we have a list including named entities and a list with associated ontology terms. This information is used to learn a classification schema using the WEKA toolset [45].

Term Annotator: Once the classifier has learned a model for each ontology term based on the set of named entities, these models are now used to annotate documents from related domains with ontology terms based on their contained named entities. To do this, the first step is to extract all named entities from the

[28] http://www.ncbi.nlm.nih.gov/pubmed

documents, e.g., by using the OSCAR framework for chemical documents. Afterwards the learned models are used to predict a set of adequate ontology terms for each document. For each assigned term, a confidence value is given indicating the probability that the term was correctly assigned to the document.

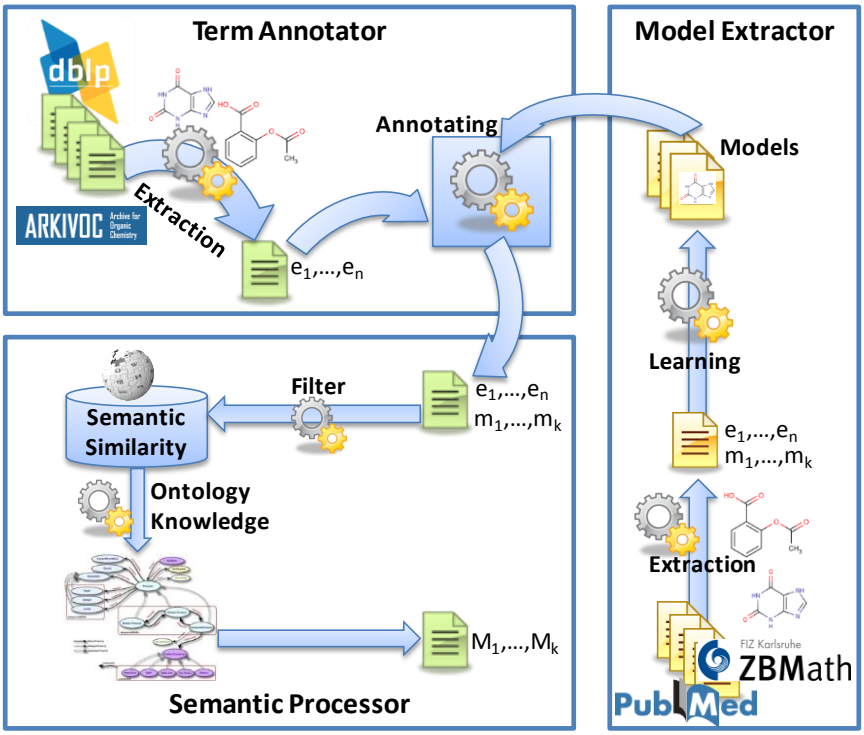


Fig. 33. System overview

Semantic Processor: The semantic processor takes the annotated documents from the annotator. The goal is to filter the set of associated terms and only keep the most relevant terms with respect to the entities included. For each entity *e* from the set of all entities *E* and for each ontology term *m* from the set of all terms *M*, we compute the semantic similarity for each pair. The relevance of an ontology term for a document is the maximum of its semantic similarity values to any entity in the document. To compute this kind of similarity, we need a knowledge base containing both, named entities as well as ontology terms.

The most prominent general knowledge base today is Wikipedia. Its usefulness for document retrieval compared to other knowledge bases, like, e.g., WordNet or Open Directory Project (ODB), was shown in [84]. We use Wikipedia as "glue" to connect the domain-specific ontology terms and the vocabulary from the target domain. As in the previous section for the context similarity, we compute the semantic similarity between a named entity and an ontology term in Wikipedia relying on the relatedness measure described in [82]:

$$relatedness(a,b) = \frac{\log(\max(|A|,|B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|,|B|))} \qquad (13)$$

where *a* and *b* are two articles, *A* and *B* are the sets of articles that link to *a*, respectively *b*, and *W* is the set of all articles in Wikipedia.

The relevance of an otology term *m* for a document *d* is defined as:

$$relevance(d, m) = \max(relatedness(m, e_n)) \qquad (14)$$

where $e_n \in E$ and *E* is the set of all named entities occurring in *d*.

Finally, we have an ontology term vector assigned to each document where the ontology terms are ranked according to their Wikipedia relevance to the document's content. The extended documents are stored in our repository. When a new document is indexed the semantic similarity between each named entity to each term from the source ontology is computed. The results are stored in a relational database.

For performing a search, the query term is extended with suitable ontology terms by the semantic processor. For result set ranking the Dice similarity based on the sets of assigned ontology terms is computed,

$$D_{sim} = \frac{2*|D_m \cap Q_m|}{|D_m| + |Q_m|} \qquad (15)$$

where Q_m is the set of assigned ontology terms for the query entity and D_m the set for the respective document. Finally, the ranked document set is delivered to the user.

Example from chemistry: The chemical domain offers access to only a few and mostly highly specialized controlled vocabularies, like, for example, *Chemical Entities of Biological Interest* (ChEBI [46]). Therefore, the key idea is to aggregate all the knowledge about chemical entities available in ontologies from other, but related domains. For instance, while the huge collection of MeSH-annotated MEDLINE documents mainly focuses on illnesses, it still relates them to drugs, i.e. chemical entities. Extensive discussions with domain specialists from different areas of chemistry showed that MeSH terms to some degree can be useful for describing properties of chemical entities. We thus use chemical entities occurring in MEDLINE documents to learn the associated MeSH terms.

Considering a synthetic chemist specialized in the synthesis of organic compounds of pharmalogical interests named Frank. Frank is searching for documents containing information from the class of compounds called synthetic cannabinoids. As a synthetic chemist, he may be looking for compounds of the naphthoylindole family acting as analgesics. During his research he finds the substance *1-pentyl-3-(1-naphthoyl)indole*, a full agonist at both the CB_1 and CB_2 cannabinoid receptors, with some selectivity for CB_2. Now, to complete his work he is especially interested in documents describing synthetic methods for the preparation and isolation of these compounds as well as possible derivates, on lab scale with highest possible yield. Moreover, also older documents might be relevant as they often contain processes that are not covered by expensive to acquire patents.

He submits the query q to our system. The query is handed on to the semantic processor which extends it with suitable MeSH terms. Please note, since Frank is a chemist only MeSH terms are used that are from the chemical sub-trees of the MeSH ontology. The extended query q_c is used for document retrieval. Here, a Boolean search is accomplished, meaning that all documents including the original query term q are retrieved. The extended query q_c is used to rank the documents according to the desired context (in this case chemistry).

4.3.2. Evaluation

For the evaluation of our approach, we used different document collections. For MeSH annotated biomedical documents, we took around 120,000 documents from PubMed Central[29], which is a free fulltext archive of biomedical and life sciences journal literature at the U.S. National Institutes of Health's National Library of Medicine (NIH/NLM).

Furthermore, for the chemical domain, we used 2700 documents from the journal Archive for Organic Chemistry (ARKIVOC), which is one of the most renowned open access sources for organic chemistry. To specifically focus on different contexts, we took around 100 manually curated documents from the Beilstein Journal of Organic Chemistry (BJOC), which is an international, peer-reviewed open-access journal dealing with all aspects of *organic chemistry*. Furthermore, we curated around 130 documents from the Eurasian Journal of Analytic Chemistry (EJAC), which focuses on all aspects of *analytical chemistry* related with analytical methods, new instruments and reagents.

We performed the following experiments:

- First, we evaluated whether a simple query expansion is already useful for entity centric search. We compared the term distributions of the EJAC and BJOC journal that are focused on different working fields: organic and analytical chemistry. Furthermore, we let domain experts define sets of context terms for both working fields. In addition, we also tried a statistical approach computing term-to-term co-occurrences for query expansion. Comparing the results using query expansion, we can state that it is not a suitable choice for enabling context-driven retrieval in chemistry.
- In the second experiment, we analyzed whether cross-domain knowledge can be useful for annotating chemical entities. We used the MeSH ontology to annotate chemical entities and discussed the results with domain experts. From the chemist's point of view, the associated MeSH terms are comprehensible and quite useful giving insights on the chemical properties as well as applications scopes. Please note, that this experiment has more anecdotic character to give the reader an illustrative example of the annotated MeSH terms for a given chemical entity.

[29] http://www.ncbi.nlm.nih.gov/pmc

- In the third experiment, we trained different classifiers to predict MeSH terms based on the chemical entities in a document. Our evaluation with a precision /recall analysis shows that it is indeed possible to predict MeSH terms using chemical entities.
- In the fourth experiment, we use the learned classification models to annotate chemical documents with MeSH terms. Comparing different classifier confidence thresholds, we present a semantic extension using Wikipedia semantic similarity to filter out irrelevant MeSH terms for chemistry.

Using Query Expansion for Contextual Search

The traditional way of searching for documents related to a specific context is to use query expansion. The user enters a query term and some context keywords. All documents containing both terms are returned. We did an experiment analyzing the word distribution of two chemical journals from different chemical working fields: organic chemistry (BJOC journal) and analytic chemistry (EJAC journal). If both collections use totally different terminology a query expansion should work to distinguish the documents. We used Apache Lucene[30] to index the documents. For the EJAC journal 55350 and for BJOC 44187 terms have been indexed. The overlap is indeed just 9012 terms. Since the overlap between the two collections is quite small, it seems that query expansion should work fine. However, if we take a closer look at how often the different terms occur in the collections, we immediately see that the terms occurring in only one collection are very rare (see **Fig. 34**).

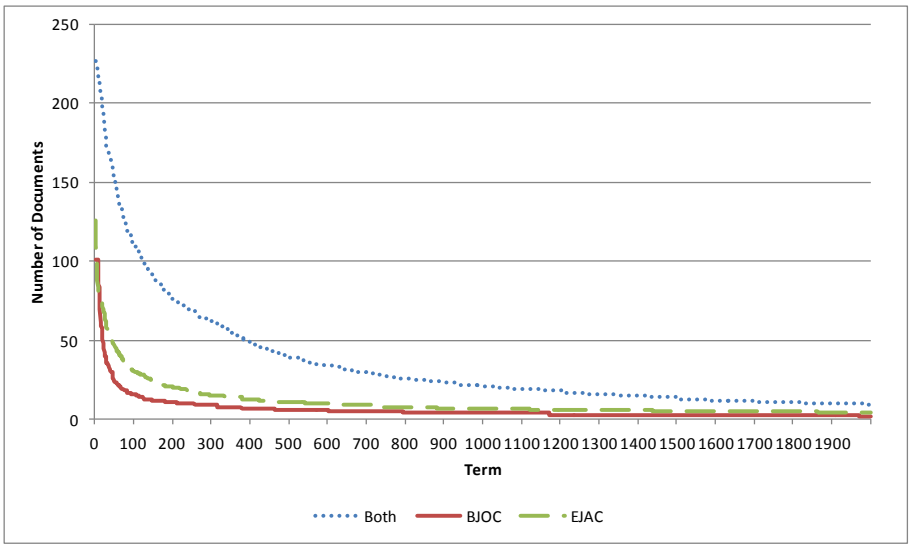


Fig. 34. Comparing term distributions of different document collections

In contrast, the terms occurring in both collections are very frequent. The top-200 terms from EJAC and BJOC occur 12,787 times in the documents. Considering

[30] http://lucene.apache.org/core

terms occurring in both collections, the top-200 terms occur in more than 25,000 times in the documents. This leads to the assumption that query expansion is no suitable choice to distinguish documents from both collections.

To prove this statement, we did a precision/recall analysis. As document collection, we used the 2700 documents from our ARKIVOC collection. Please note that these documents are from the same chemical sub-field as the BJOC collection: organic chemistry. For each of these documents, we extracted all chemical entities using the OSCAR framework. Since relevance can only be assessed manually by domain experts (making it a very expensive process), we performed the precision/recall analysis only on a subset of documents (still about 10% of the entire collection). To choose a *representative* subset, we analyzed the number of occurrences of individual chemical entities in the document collection. **Fig. 35** shows the distribution of the 20000 most often occurring chemical entities.

Since it is not sensible to choose entities for evaluation that either occur in almost all documents or are extremely rare, we chose our query entities for evaluation only from entities occurring in less than 100, but more than 20 documents (see the shaded area in **Fig. 35**). We retrieved all documents matching the queries and randomly chose a subset of 10%. From these documents, we randomly selected a total of 5% of the occurring entities resulting in 22 textual query terms.

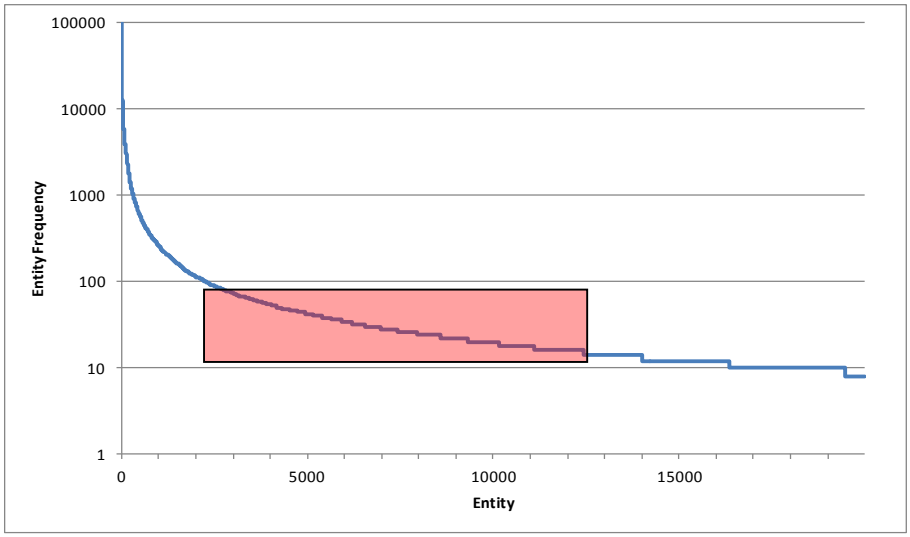


Fig. 35. Entity distribution in collection

For a first experiment, we also added the EJAC documents to our set and computed a Lucene index. Here, we were interested in receiving all documents from the area of organic chemistry (ARKIVOC journal). All documents from the ARKIVOC journal containing the query term are marked as relevant. Of course, for a simple Boolean search without any context restriction all documents containing the query term have been found. But, there are also a lot of irrelevant documents for the context "*organic chemistry*" leading to a low precision of only 31.6%.

To enhance the precision, we used a statistical query expansion method to define context terms. Since we are interested in documents for organic chemistry, we computed a term-to-term co-occurrence matrix based on the ARKIVOC document subset. For each query term, we retrieved the top-10 most co-occurring terms for the context "*organic chemistry*". The position of the term in the document is also taken into account, meaning that terms that are closer to the query term will get a higher score. Furthermore, we used popularity thresholds defining a required minimum and maximum popularity. Terms not fulfilling these thresholds are not used as context terms. Finally, the query is expanded with the top-10 co-occurring terms using the same query model as in Chapter 4.2.4: Let $C=\{q_c,c_1,\ldots,c_n\}$ be the set including q_c and all expanded terms. The expanded context query is formulated as q_c *OR* c_1 *OR* … *OR* c_n, meaning all documents are returned containing q_c or any of the expanded terms. This expansion leads to a small increase of the precision to 34.1%, but to a high decrease of the recall to 50.57%. The reason for the decrease of the recall is that the quality of the automatically generated context terms is not sufficient. Have of the relevant documents do not contain any of the most co-occurring terms.

We also did a second experiment, where the relevance assessment was done manually by domain experts. The experts considered all retrieved documents with respect to each query and judged the relevance in a binary fashion. As in our use case, we chose the sub-domain of "*synthesis chemistry*" for context search. The search is performed using a Lucene index on the documents. The average precision for a search using only the query terms is 17.1%, which is very low.

To enhance the precision the experts defined a set of typical context terms, which are used for query expansion, like, e.g., synthesis, reduction, reaction, catalysis or oxidation. But, using the combination of query term and context terms the precision actually decreased to 14.42%. Also the recall decreased to 45.1% meaning we miss relevant documents due to the context restriction. To ensure that the reason for the bad results is not the manual selection of the context terms, we also used a statistical approach for context term selection. Here, we computed the term-to-term co-occurrence matrix based on all relevant documents (133 in total). But, as before, we could not get satisfying results. The precision increases compared to the manually selected context terms up to 23.1%, but the recall decreases to 41.4%.

These results indeed proved that a simple query expansion is not useful for context-driven searches in chemistry. Therefore, we can state an urgent need for additional annotations for the documents to enable context-driven searches.

Are MeSH Terms Useful for Describing Chemical Entities?
Please note, that this experiment has more anecdotic character to give the reader an illustrative example of the annotated MeSH terms for a given chemical entity. While analyzing the MeSH vocabulary with domain experts, we found out that many of the included terms are also useful for describing chemical documents. Whole subtrees of the ontology deal with chemical substances and general terminology. For example, 2964 nodes are listed in the sub-tree for 'Organic Chemicals'.

Fig. 36 shows an extract of the MeSH ontology dealing with chemical terminology for the node 'Chemistry'. The tree shows that there are different sub-nodes that represent different concepts from the chemical domain, like, e.g., organic chemistry or analytic chemistry.

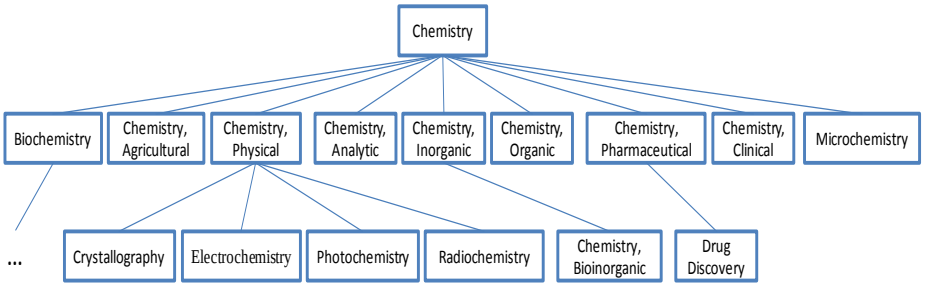


Fig. 36. Extract of MeSH ontology for term 'Chemistry'

In a first experiment, we tried to find out if the used terminology in MeSH is comprehensible for experts from the chemical domain. Therefore, we took the extracted chemical entities from our ARKIVOC collection and searched for them in our PubMed Central collection. In total, we have 164817 unique chemical entity names in the ARKIVOC collection. 151287 (91.8 %) of them can also be found in PubMed Central.

To evaluate the MeSH vocabulary we annotated each chemical entity with a set of MeSH terms. We searched for the respective entity name in the titles and abstracts of the PubMed documents. If the name is found in the document, the document's MeSH terms are added to the entity's term set. We did not use the document's fulltext, because if the entity occurs in title or abstract, it should be more important for the document's context as if it occurs just somewhere in the fulltext. For each entity, we created a tag cloud including all associated MeSH terms. As usual, the font size within the clouds is defined by the number of occurrences (i.e. the significance) of the respective term. We showed the tag clouds to domain experts and discussed, if they can associate the used terminology in the cloud with the chemical substance. From the experts' point of view, the used terminology was comprehensive and while it contained some unrelated information, most of the terms were considered quite useful. To give an illustrative example, **Fig. 37** shows the MeSH term cloud for the chemical entity *Formaldehyde*.

For a long time, Formaldehyde was used in chipboards as agglutinant, respectively binding material. Due to its cancer-causing properties its evaporation leads to a contamination of the indoor air. Therefore, while not chemically relevant in a narrow sense terms like 'Air Pollution', 'Air Pollutants' or 'Indoor' occur prominently in the tag cloud. Terms like 'Carcinoma', 'DNA', or 'Neoplasmas' refer to the carcinogen effect that strongly confined the use of Formaldehyde. There are a lot of terms in the cloud dealing with the subject of cancer or biochemical processes. 'Receptors' indicates the cancer impact focused on biochemical aspects. Furthermore, the term

'Disinfectants' is one of its original fields of application, but still very useful for the individual chemists' context.

Fig. 37. MeSH term-cloud for Formaldehyde

Predicting MeSH Terms Using Chemical Entities

In this experiment, we aim at learning classification models to assign MeSH terms to documents based on their chemical entities. We tried different classifiers using the WEKA framework. First of all, we needed to find out if chemical entities can be used to predict MeSH terms at all. For evaluation, we took the 120,000 documents from the PubMed Central collection. Again, we used the OSCAR framework to automatically extract all chemical entities. From the set, around 114,000 documents include at least one chemical entity and could therefore be used for classifying. In total, we found 151,287 unique chemical entities in the collection.

Of course, every document may have several MeSH terms. The problem is that WEKA does not support this kind of multi classes. Hence, it is necessary to train several classifiers: One classifier for each MeSH term. Furthermore, it is important to get enough positive instances for each class to train the classifier. Therefore, we only used terms as classes that are included in at least 10 documents. Our goal is to predict the classes based on the chemical entities. Thus, we have for each MeSH

term a file containing all chemical entities as attributes (around 150,000) and the respective MeSH term as class attribute. The instances are the documents represented in a sparse vector format, where each dimension specifies the occurrence of the respective attribute. We did not choose *all* instances randomly, because then, due to our large dataset, the probability that most of the instances do not belong to the class is high. That would mean that during testing the probability is high that the classifier will not assign this class to an instance. Therefore, for each class we took all documents belonging to the class (positive examples) and randomly choose the same number of documents not belonging to the class (negative examples). Before training the classifier, we used the *RemoveUseless* filter from Weka to remove all irrelevant entities for the respective class. In total, we trained 8381 different classes.

We tried three different classifiers and compared their results in a precision/recall analysis. For all classifiers we used the default options and 10-fold cross-validation. The results are shown in **Table 14**.

Table 14. Average precision and recall of different classifiers (in %)

Classifier	Class yes		Class no	
	Precision	Recall	Precision	Recall
naïve Bayes	79.62	77.98	**79.67**	78.50
C4.5 (J48)	79.10	66.99	71.50	**81.49**
SVM (SMO)	**79.72**	**80.00**	76.91	78.87

The labels 'class yes' and 'class no' mean that the classifier predicts that a document has, respectively has not, the given class. The best classifier is the SVM having precision and recall values of around 80% for all cases. The SVM implementation in WEKA is named SMO and implements the sequential minimal optimization algorithm for training a support vector classifier, see [85] for details. The results show that it is possible to use chemical entities for assigning MeSH terms to documents.

Annotating Chemical Documents with MeSH Terms

In this experiment, we assigned MeSH terms to chemical documents and assessed their usefulness. We used the SVM classifier to annotate each of the 2700 documents of our ARKIVOC collection. The classifier takes all entities from each document and applies all learned models. In total, we have around 8000 different classes. **Fig. 38** shows the number of associated MeSH terms for each document. In average 3316 terms are assigned to a document. If the classifier decides to assign a term (class) to the document, also a confidence value is computed.

To know which terms are more related to chemistry than others, we analyzed the MeSH ontology with domain experts and figured out important parts of the ontology for the domain of chemistry. The MeSH ontology consists of 19 main categories ranging from 'Anatomy' to 'Geographical Location'. Of course, not all of

them are relevant from the chemist's point of view. From the 19 main categories we identified the 'Chemicals and Drugs' category to be of special interest for chemists. This category contains 20,249 sub-categories covering, for example, a lot of different organic and inorganic chemicals. Another interesting sub-tree containing more general terms, called 'Chemistry', can be found under the 'Natural Science Disciplines' node in the 'Disciplines and Occupations' category (see **Fig. 36**).

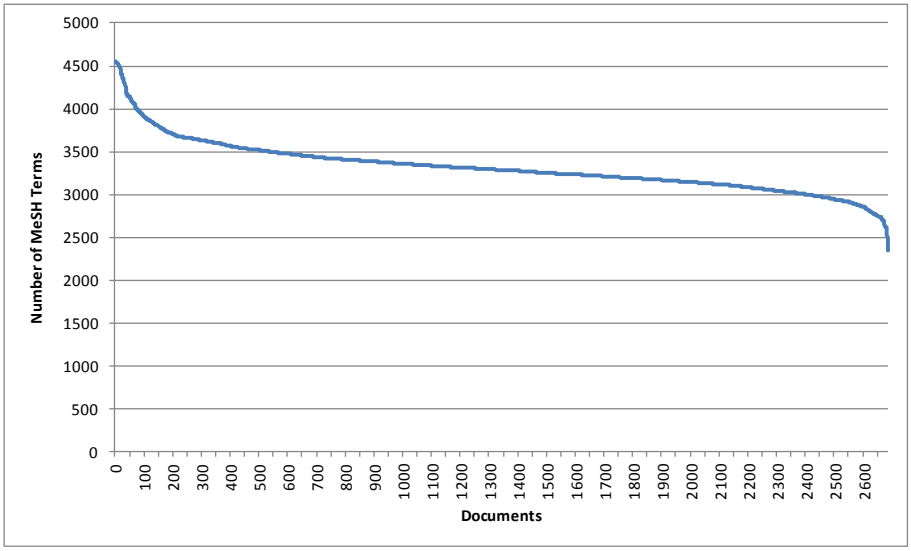


Fig. 38. Number of assigned MeSH terms per document

To evaluate the usefulness of our approach we have to determine the quality of the assigned terms. Therefore, we defined that all terms from the chemical sub-trees are relevant. We took the assigned MeSH-terms from each ARKIVOC document and ranked them according to their confidence value. Then we took the top-k terms and computed the precision for varying k's (precision@k).

Fig. 39 shows the average precision@k for different confidence thresholds. A confidence threshold of, e.g., 0.5 means that each assigned term has *at least* a confidence of 0.5. The precision values are low for almost all confidence thresholds (around 37%). The highest value is reached for a confidence threshold of 0.3 for the top-10 MeSH terms. The average precision is 38% meaning that from 10 assigned terms only 4 are relevant for the area of chemistry. The problem is that the confidence value does not describe how the term is semantically related to the document. It only says to what percentage the classifier is sure that the term has to be assigned to the document. To further enhance the quality of the assigned terms we need a semantic filter. Therefore, we used Wikipedia to compute the relevance of a MeSH-term for the respective document. The relevance is defined as the maximum semantic similarity of an assigned MeSH-term compared to each chemical entity occurring in the document. Again, we did a precision@k evaluation, this time varying the relevance threshold.

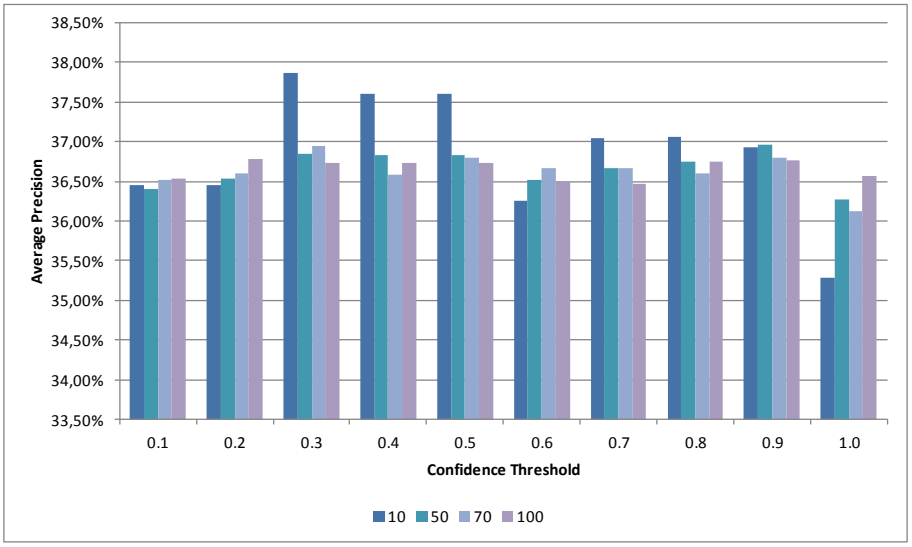


Fig. 39. Average precision for varying confidence thresholds for top-k MeSH terms

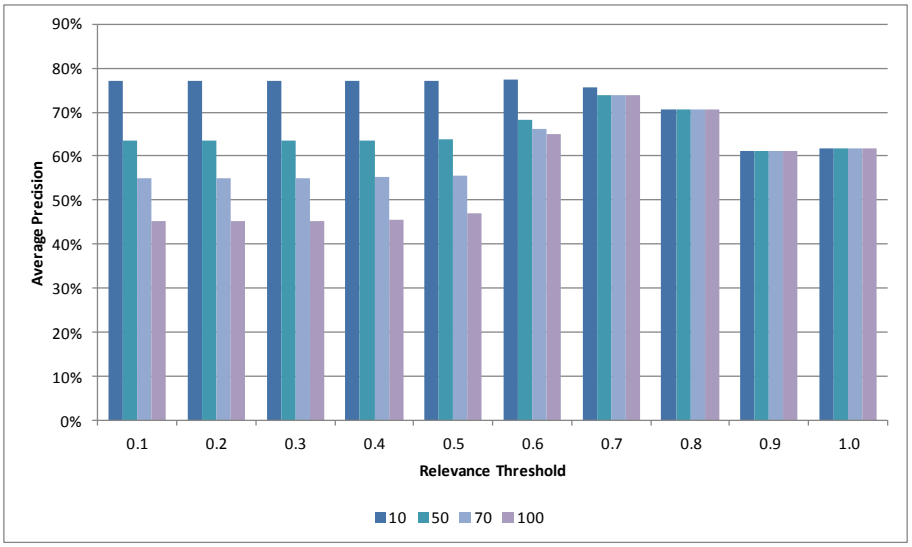


Fig. 40. Average precision for varying Wikipedia relevance thresholds for top-k MeSH terms

Fig. 40 shows the results of the average precision@k evaluation for varying Wikipedia relevance thresholds. The results show that the average precision is much better using the Wikipedia relevance. For the top-10 assigned terms the best average precision (78%) is reached for a relevance threshold of 0.6. However, for the top-50 to top-100 terms the precision drops to around 65%. Regarding all top-k terms, a threshold of 0.7 retrieves the best results with an average precision of always at least 74%.

This experiment has shown that using the knowledge from Wikipedia or similar sources can dramatically increase the quality of the assigned MeSH-terms. The combination of MeSH-terms and Wikipedia seems to be quite useful to enrich chemical documents.

4.3.3. Retrieval Based on Cross-Domain Knowledge

In this section, we show the usefulness of the cross-domain annotations for enriched document retrieval. For our experiments, we used the same datasets as introduced in Chapter 4.3.2. In addition, to prove the general usefulness of our approach, we also did experiments with document collections from other domains as chemistry. We took the Zentrallblatt Math (ZM) document repository containing 3 million documents. Each document is annotated with several terms from the MSC taxonomy. Furthermore, we took the DBLP computer science document repository containing 638000 documents. Since these documents lack suitable annotations, we use cross-domain knowledge from the ZM documents to improve the retrieval quality.

We did two experiments to show the usefulness of the annotated cross-domain ontology terms for document retrieval.

- To show that using the annotated documents context-driven searches are possible, we compare the results to a BM25 ranking and an enhanced baseline taking Wikipedia category information into account. The results indeed prove that our approach promise to dramatically increase the user's search experiences.
- Furthermore, we prove the general usefulness of our approach. We enrich documents from the area of computer science with terms from the related domain of mathematics and evaluate the retrieval results.

Using MeSH for Chemical Document Retrieval

In this experiment, we analyzed whether the assigned cross-domain MeSH terms really lead to suitable improvements for chemical document retrieval. As document sets, we used our PubMed Central (PMC), ARKIVOC and BJOC collections. The documents from ARKIVOC and BJOC are all from the area of organic chemistry and are therefore closely related. In total, we got 120,000 documents. We randomly chose 25 query terms out of all chemical entities from our collection. We are interested in documents containing the respective query entity in the context of organic chemistry. For each query, we took 50 documents from the organic chemical journals and 50 documents from PMC. We only took documents where the respective query entity occurs in title or abstract. The relevance was assessed manually by domain experts. For each of these queries, we computed the semantic similarity to each of our learned MeSH terms using Wikipedia. We assigned all MeSH terms with a relevance threshold of more than 0.7 to the respective query term. Since we are interested in retrieving all documents in the context of organic chemistry, we filtered the assigned MeSH terms to only use terms from the respective sub-tree of the MeSH ontology.

All documents in our set are already annotated with MeSH terms. The PMC documents in our collection have in average around 10 MeSH terms. Therefore, we also used the top-10 terms for our chemical documents. The terms are ordered by the Wikipedia relevance score. For performing a search, all documents containing the respective query term are retrieved. For result set ranking, we computed the Dice similarity on the sets of assigned MeSH terms.

To evaluate if the annotation of MeSH terms leads to better retrieval results we compared the results to two different baselines. The first baseline uses the BM25 ranking model with standard parameters. We searched for the 25 query terms using a Lucene fulltext index without additional MeSH terms for the chemical documents. Furthermore, we also compared our approach to a Wikipedia category baseline to evaluate the retrieval improvement of the semantic processor. As described in Chapter 5.2, we annotated each document with Wikipedia categories based on its contained chemical entities. Also all query entities are annotated with Wikipedia categories. Again all documents containing the query entity are retrieved and ordered using Dice similarity based on the annotated categories.

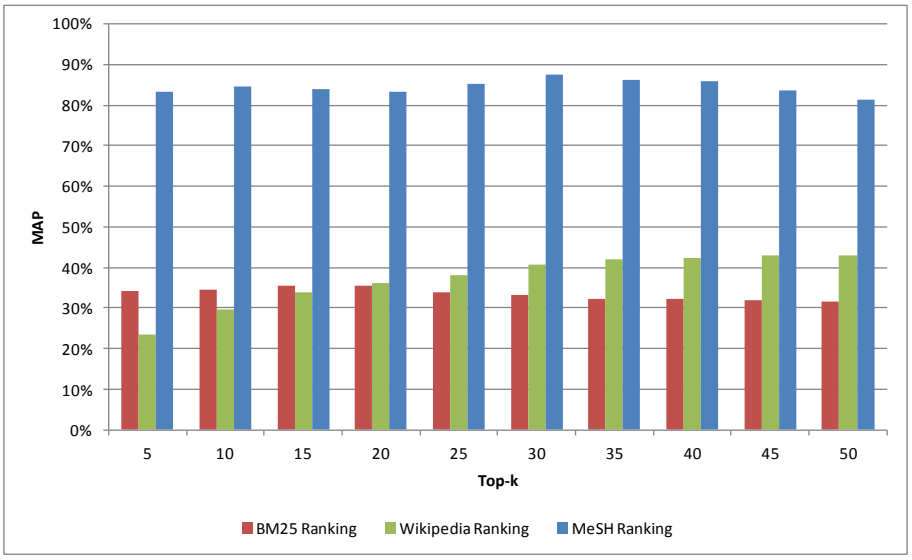


Fig. 41. MAP for top-k documents

We compared our ranking to the results of the BM25 ranking and the Wikipedia category baseline. To compare the different rankings, we computed the Mean Average Precision (MAP) for the top-k documents over all queries (see **Fig. 41**). For the BM25 ranking the precision values are around 35% with the highest value of 35.54% for the top-20 documents. The Wikipedia ranking has a low precision value of 23.5% for the top-5, which increases to 43.2% for the top-45 documents. Using the MeSH annotations the average precision can be dramatically improved. For our MeSH ranking the precision values are almost constant around 83% with the highest value of 87% for the top-30 documents.

These results show that using the knowledge about chemical entities from other domains for extending chemical documents promises a high increase of the retrieval quality for domain experts. Without additional annotations, the top-k result sets include more than 60% of irrelevant hits. With Wikipedia annotations, this number can still be decreased to 50%. Using semantically enriched documents only 15% of the retrieved results are irrelevant.

Enriched Retrieval for Computer Science

In this experiment, we prove the general usefulness of our approach. We took documents from the ZM repository, where each document is annotated with several terms from the MSC taxonomy. While analyzing the taxonomy, we found out that a whole sub-tree is relevant for the related domain of computer science. Therefore, we took the DBLP document repository containing 638,000 documents from computer science. Since these documents lack suitable annotations, we use cross-domain knowledge from the ZM documents to improve the retrieval quality.

We extracted named entities from the ZM documents and trained a SVM classifier to learn the MSC classes. The entity extraction was done using the Wikipedia Miner, which annotated all entities matching to Wikipedia articles. We also extracted named entities from the DBLP documents and associated MSC classes based on the learned classification models. The assigned MSC classes are filtered using the semantic processor. Finally, the usefulness of the annotations is evaluated in a document retrieval experiment.

Therefore, we randomly choose 30 query entities and took 150 documents containing these entities from DBLP and 150 documents from ZM. The relevance of each document for each query was manually judged by a group of 10 domain experts. All experts are Ph.D. students or postdoctoral researchers from the field of computer science. The goal is to find documents containing the query term, which are relevant for the context computer science. As described for the MeSH experiments the query term is associated with terms from the MSC taxonomy. Since the context is computer science, the terms are filtered to those from the respective sub-tree. Again, we compared against the BM25 and the Wikipedia categories baseline. **Fig. 42** shows the results for the top-k retrieved documents using MAP.

Interestingly, the results for the BM25 ranking are better than in the chemical domain. The reason is that the query terms in computer science are more general than chemical entities leading to better retrieval precision also for fulltext searches. Nevertheless, the cross-domain ranking using MSC classes outperforms both baselines. The highest MAP of 85.9% is reached for the top-5 documents.

This experiment proved that cross-domain knowledge from related domains is very useful to improve the retrieval quality. However, it is important to also filter the annotated cross-domain terms to ensure that they are semantically related to the document's context. Therefore, it is important to use a general knowledge base,

like, e.g., Wikipedia, as 'glue' to connect the domain-specific ontology terms to the vocabulary used in the other domain.

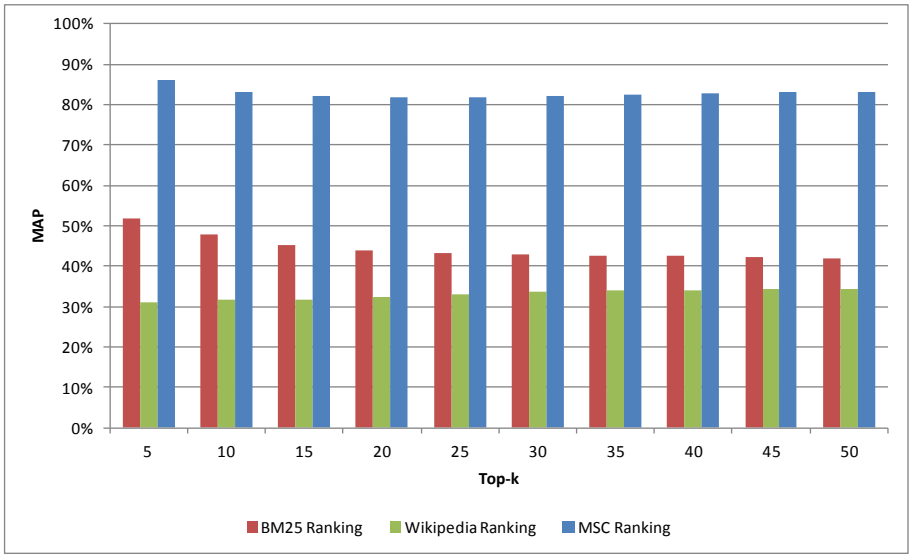


Fig. 42. MAP for top-k documents in computer science

4.4. Using Wikipedia to Overcome the Vocabulary Problem for Contextual Queries

The annotated terms will not help for contextual queries if the user uses different vocabulary than provided by the annotated terms. In this section, we explain our approach to semantically enrich documents to overcome the vocabulary problem enabling contextual queries. Our basic idea is to extract important terms from documents and use Wikipedia to compute the semantic similarity between these terms. Also previous work used Wikipedia to help users finding relevant query terms and interactively guide them on their search [86]. Since Wikipedia uses the wisdom of the crowds, which has been proven to provide tremendous quality [87], the contained knowledge is growing fast and updated regularly. **Fig. 43** gives an overview of our approach.

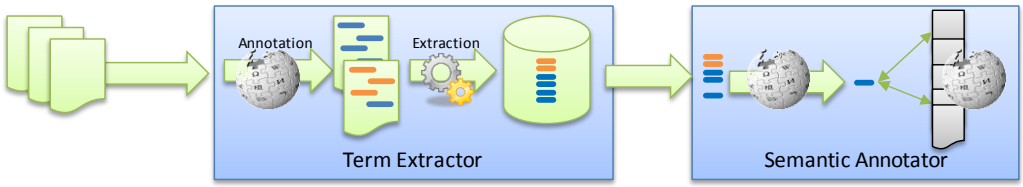


Fig. 43. Workflow overview

The architecture is composed of two basic components. The term extractor is responsible for annotating and extracting important terms from the documents. For annotating the documents, we use the Wikipedia Miner toolkit. The main purpose

of the Wikipedia Miner is to annotate a given fulltext in the same way a human would annotate a Wikipedia article. The methods are based on a machine learning approach, which is used to identify relevant terms and links them to Wikipedia. The approach is two-folded in the way that the first task is to disambiguate the terms, which occur in a given text, and the second task is to check whether the detected terms are useful links to Wikipedia articles.

The extracted terms are further processed by the semantic annotator. For each term its associated Wikipedia categories, and its in- and out-links are extracted. These features are used for computing the semantic similarity between different terms. The measures used for calculating the feature similarities have been introduced in Chapter 4.2.1.

We evaluated this approach in a retrieval scenario. **Fig. 44** gives an overview of the retrieval workflow. The user enters a query and submits it to our system. For retrieving a ranked list of relevant documents, the system is composed of two components: the semantic annotator and the ranking engine.

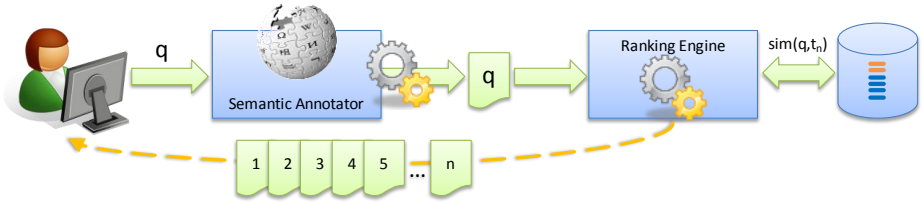


Fig. 44. Retrieval workflow

The query term *q* is analyzed by the semantic annotator, which enriches it with the different similarity features extracted from Wikipedia. The ranking engine receives the enriched query term and creates a ranked list containing all other terms. In case *q* is already known in our system, the semantic similarity ranking is directly received from the relational database. Otherwise, it is necessary to compute the similarity to all terms known to the system. For our repository, containing 34324 different terms, the similarity computation for an unknown term took less than three seconds. Finally, the documents are ranked according to the similarity values of their contained terms. The relevance of a document *d* to a query *q* is computed as follows:

$$rel(\mathrm{q}, d) = \frac{\left(\sum_{\mathrm{t} \in \mathrm{q}} \sum_{x \in T_d} \frac{sim(t_x, \mathrm{t}) * \tau_{\mathrm{t_x}}}{|T_d|} * \frac{\omega_{\mathrm{t_x}}}{\Omega_{\mathrm{t_x}}}\right)}{|q|} \qquad (16)$$

where T_d is the set of all terms included in *d* and τ is a boosting factor to give terms occurring in the document's title a higher weight. Each query *q* can consist of several terms *t*. Furthermore, ω denotes the number of times a term occurs in a document. This value is normalized by the number of times the term occurs in the whole collection, denoted by Ω. Finally, the score is normalized by the number of terms in *q*.

As document repository, we use 122,640 documents from the PUBMED Central corpus, which is part of the MEDLINE repository. Each document in this set is manually annotated with several terms from the MeSH ontology, which offers a controlled vocabulary for indexing and retrieval purposes. These terms are abstract concepts describing the general context of the respective document. Therefore, we also use MeSH terms as query terms in our experiments. To find a set of suitable query terms, we analyzed the distribution of the MeSH terms in our document collection. As possible query terms we considered all terms occurring in less than 1000 but more than 10 documents. From this set, we randomly choose 80 query terms, which also occur in Wikipedia. As document set for the experiments, we used all documents that have been annotated with at least one of these query terms. The MeSH annotation is done manually by domain experts resulting in high quality. Therefore, for our evaluations we considered all documents annotated with the respective MeSH term as relevant hits. In total, our set contains 10791 documents.

4.4.1. Comparing to Different Baselines: Lucene Index, Statistical Query Expansion, and Latent Semantic Analysis

In this experiment, we searched for all query terms in the documents' fulltext. Therefore, we created a Lucene fulltext index including all documents from our subset. To analyze the retrieval quality, we considered all documents annotated with the respective MeSH term as relevant hits. The documents have been ranked according to the BM25 ranking model using standard parameters. As evaluation measure, we computed the Mean Average Precision (MAP) and the average recall over all queries. Our experiment results in a MAP of 31.53% and an average recall of 37%.

To enhance the MAP and the recall, we also used a statistical query expansion method. We computed the term-to-term co-occurrence matrix based on the documents of our subset. The position of the term in the document is also taken into account, meaning two terms that are close together will get a higher score. Furthermore, we used popularity thresholds defining a required minimum and maximum popularity. Terms not fulfilling these thresholds are also not used as expansion terms. We used the following retrieval model: Let q be the query term and $C=\{c_1,c_2,\ldots,c_n\}$ the set of all expansion terms. For the expanded query, the queries are formulated as $q\ OR\ c_1\ OR\ c_2\ OR\ \ldots\ OR\ c_n$, meaning all documents are returned containing the query term or at least one expansion term. Finally, the query is expanded with the top-k co-occurring terms. **Fig. 45** shows the results for the top-k expansion terms.

The best MAP of 40.28% is reached for the top-21 expansion terms. As expected, the more terms are added to the query the higher is the recall. The maximum recall of 81.91% is reached for the top-58 terms.

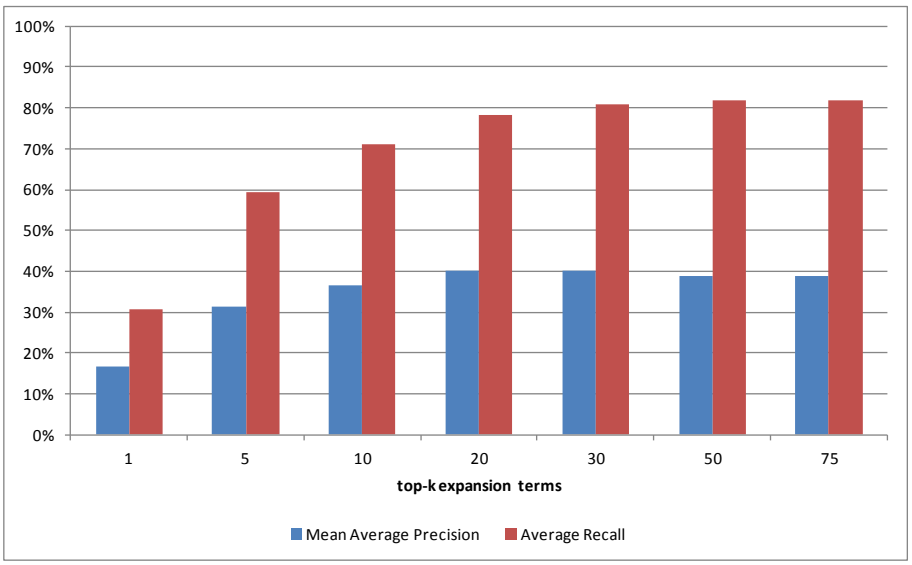


Fig. 45. MAP and average recall for the top-k expansion terms

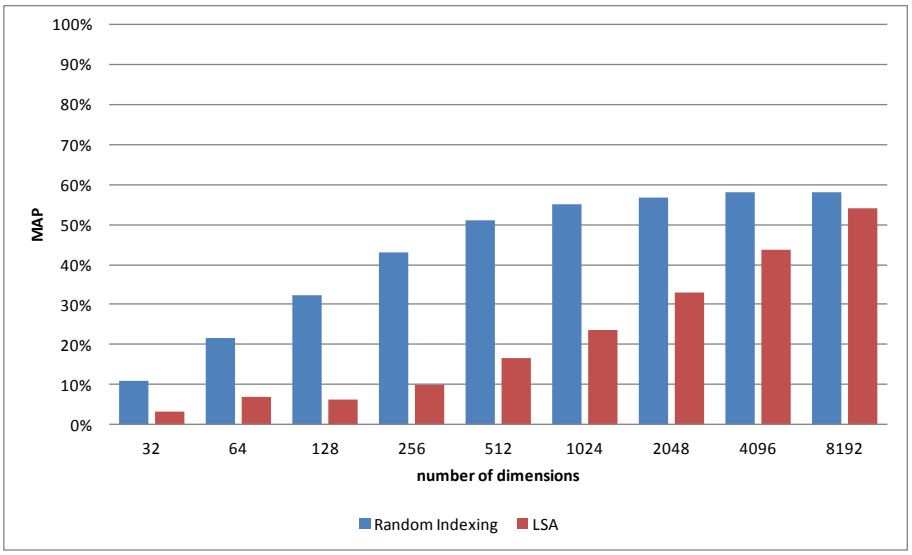


Fig. 46. MAP for Random Indexing and LSA

Beside query expansion, we also evaluated how an LSA approach would perform in this scenario. To analyze the performance of an LSA based approach, we used the Semantic Vectors[31] toolkit, which is built upon Apache Lucene. We used LSA and Random Indexing for building the vectors for our corpus. Random Indexing is an alternative approach to standard word space models, which is efficient and scalable [88]. For both methods, we used the standard parameters and varied the number of dimensions used for the vectors. We started with 32 dimensions and went up to

[31] https://code.google.com/p/semanticvectors

8192 dimensions. The resulting MAP of booth methods was continuously growing with an increase of the number of dimensions. We did not use a higher number of dimensions because of the runtime complexity and memory requirements for the resulting model.

The results are shown in **Fig. 46**. We see that the MAP based on Random Indexing is higher in all cases, reaching up to a maximum MAP of 58.2%. Using a very high number of dimensions, we archive quite similar results using LSA (54.1%).

4.4.2. Semantic Enrichment Using Wikipedia

In this experiment, we evaluate the usefulness of our approach for contextual queries. For each document and each annotated term, a confidence value has been computed describing the reliance of the assignment between Wikipedia article and term. We did two main experiments analyzing the influence of the confidence value. In the first one, we computed the MAP using different confidence thresholds. In this experiment, for computing the relevance of a document to a query term only the assigned terms having a higher confidence value than the threshold are used. In the second main experiment, we ordered the assigned terms for each document by their confidence values. For the relevance computation, only the top-k terms for each document are used. Furthermore, in both experiments, we also analyzed the influence of giving terms occurring in the document's title a higher weight. In addition, we also considered the number of times the term appears in the document in the ranking function. To do not prefer frequently used terms that are not descriptive for the respective document, we normalized this value by the number of times the term occurs in the whole collection. Since our method computes the relevance of a query to all documents in our set, the recall is always 100% and therefore not meaningful at all. To evaluate the different rankings and compare them to the baseline approaches we compute MAP.

Fig. 47 shows the results for the confidence threshold experiment. A confidence threshold of zero means that all terms have been used for the relevance computation. The results show that giving the terms occurring in the documents' title a higher score leads to a decrease of the MAP. We only show the results for a title boost factor of two, meaning the title terms are twice as important as other terms. In our experiments, we varied the boosting factor from one to 15. But, the higher the boosting factor the worse the results. Also the number of occurrences of a term does not lead to better overall results. The combination of title boost an occurrences leads to better results for smaller thresholds than using the features alone, but the overall best results are achieved if all terms are considered as equally important. The best MAP of 63.14% is reached for a confidence threshold of 0.1. The higher the threshold the fewer is the number of assigned terms for each document.

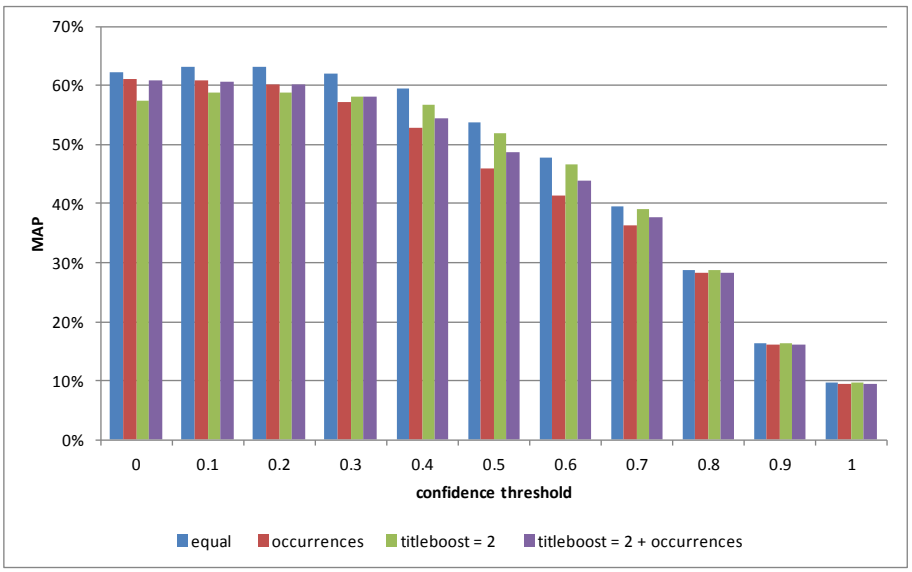


Fig. 47. MAP for varying confidence thresholds

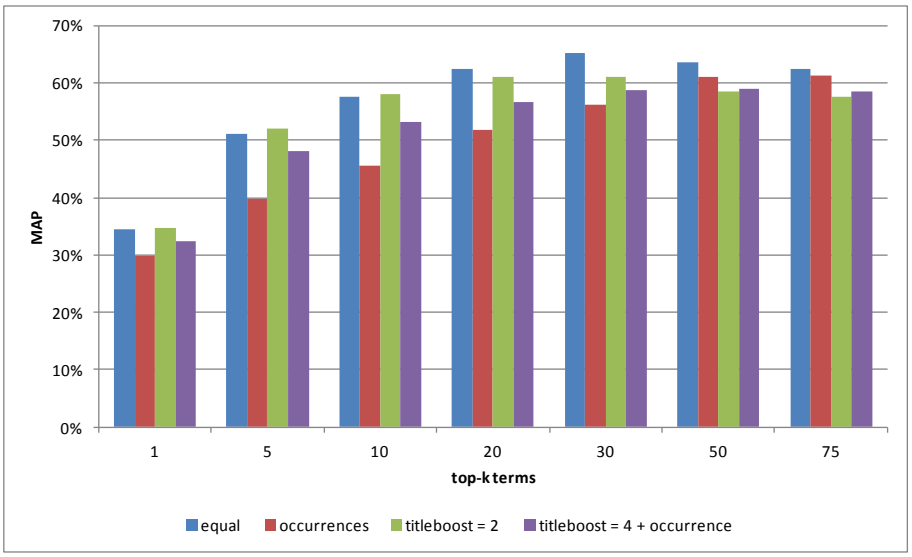


Fig. 48. MAP for top-k terms

Fig. 48 shows the evaluation results for using the documents' top-k terms. We analyzed the distribution of assigned terms for the documents in our collection. Around 10% of the documents in our collection have more than 75 terms assigned. Therefore, we computed the MAP for up to the top-75 terms. Please note, we always used all documents and only limited the number of assigned terms. As in the confidence threshold experiment, the best results are achieved if all terms are considered as equally important. Using a title boost factor or taking the number of occurrences into account does not lead to better retrieval results. The best MAP of

65.14% is reached for using the top-31 terms of each document. The MAP is slightly higher as for the confidence thresholds.

As last experiment, we analyzed the different combinations of the features used in our similarity measure. **Fig. 49** shows the results for the different combinations. This experiment shows that the categories are performing worst with a best MAP of 59% for the top-31 terms. The overall best MAP of 74.5% is reached for the top-61 terms using only the in-links feature.

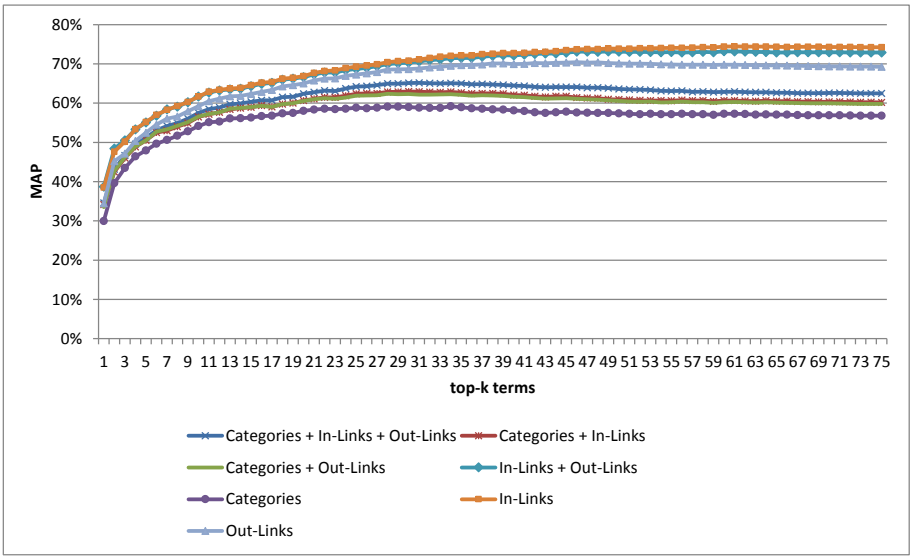


Fig. 49. Comparing MAP of different features

Overall, we showed that for conceptual queries the proposed method leads to better results than state-of-the-art retrieval models. The best baseline approach was Random Indexing achieving an MAP of 58.2%. Our approach significantly outperforms (p-value of 0.03 using a two-tailed t-test) the best baseline by achieving an MAP of 74.5%.

4.5. Conclusions

Today, entity centric search plays a major role in information gathering. However, due to the huge amount of available information, for most searches the entity itself is not enough to retrieve satisfying results. Users are usually searching for an entity in a specific context. Two entities can be very similar regarding one specific context, but the same entities might be very dissimilar in other contexts. Therefore, for digital library providers it is important to also consider this context to restrict the amount of retrieved results and increase the retrieval quality.

One important aspect to consider is the well-known vocabulary problem, i.e., users often use different query terms not matching the vocabulary of the documents. We presented an approach overcoming this problem by using external knowledge provided by Wikipedia. We took a document collection from the PubMed Central

repository and extracted the most important terms from each document. These terms are semantically enriched with features gathered from Wikipedia. Finally, the relevance of a document to a contextual query is computed resulting in a ranked retrieval list. Our evaluation has shown that our approach outperforms state-of-the-art query expansion and LSA approaches resulting in an increase of the Mean Average Precision of 58.2% for LSA to 74.5% for our approach. All results have been proven to be statistically significant (p-value of 0.03 using a two-tailed t-test). The proposed method bridges the gap between user queries and documents' fulltext by using Wikipedia for semantic enrichment.

Furthermore, we showed in this chapter that structure-based similarity measures could not retrieve suitable results for contextual searches in chemistry. Therefore, we presented two approaches enabling contextual searches in chemistry. The first uses knowledge gathered from Wikipedia. The presented similarity measure is composed of six different features extracted from the Wikipedia pages of the chemical entities. Besides Wikipedia features, like categories, in- and out-links, we also used OpenCalais to analyze the Wikipedia pages and extract additional features. The features are combined in a linear fashion and used to compute entity similarities. For context similarity, we also relied on Wikipedia and computed the semantic similarity of each chemical entity to the specific query context. Finally, entity- and context similarity are combined in one similarity measure. Our evaluations indeed showed that fingerprint-based similarity measures are not the right choice for contextual searches in chemistry. The feature-based measure relying on Wikipedia increased the retrieval results in Mean Average Precision up to 30%. Furthermore, we showed that it is possible to further increase the retrieval quality using a personalization. We analyzed which feature combination is most preferred by each chemist using user feedback. In average over all users, the Mean Average Precision increases around 9% using personalization.

Another possibility to enable contextual queries is to use ontology knowledge to annotate documents with matching context terms, like, e.g., done in the MEDLINE corpus. However, most domains cannot rely on suitable ontology knowledge. Especially in the linked open data community, many domain specific collections need manual assistance, which is hardly manageable. Considering the domain of chemistry, documents are usually missing suitable context annotations. Therefore, we presented an approach using cross-domain knowledge from the biomedical domain to learn models for annotating chemical documents with terms from the MeSH ontology. To assure that the annotated terms are semantically related to the documents' context, we used a general knowledge base, i.e., Wikipedia, to filter out all unrelated terms by computing the semantic similarity between each term and the document's named entities. We showed in a document retrieval scenario that using the annotations the retrieval quality is highly increased. Compared to a BM25 and a Wikipedia category baseline the retrieval performance in Mean Average Precision is increased more than 40%. We also proved the generalizability of our approach by annotating

documents from computer science with terms from the related domain of mathematics. Also here, the retrieval quality in Mean Average Precision is increased up to 40%. To assure that the associated terms are semantically related to the document's content, it is important to use a general knowledge base, like, e.g., Wikipedia, as glue to connect the domain specific ontology terms to the vocabulary used in the other domain.

Chapter 5

Comprehensible Representations of Retrieval Results

In the last chapter, we showed how to consider the context a user is interested in to improve the quality of the retrieval results. However, in most cases a user query returns a large set of matching results that are somehow related to the query term. However, it has often been shown that consumers usually only examine the first 10 to 20 results, respectively the first two sites of the result set[32]. Of course, the result set is ordered following a complex ranking system, which indeed is not really transparent for the user. The question is why are these pages really marked as relevant regarding the query term?

Example: Let us consider a practitioner from the field of chemistry who is searching for the query term *methoxybenzene*. Using a Web search engine the result set contains a lot of different chemical substances. The reason is that the word *methoxybenzene* is included in many different chemical substances. For example, documents including *1-Allyloxy-2-methoxybenzene*, as well as documents including *1-Fluoro-4-Methoxybenzene* are retrieved, even if the two substances have totally different properties. To ease the access to the related documents and to give a good overview of the document's content it would be useful to assign more general concepts to each retrieved document.

To give users a certain feeling, in most platforms the result lists are accompanied by snippets where the query term is highlighted. However, it is usually still not possible to get a good overview of the general topics relevant for the query from these snippets. For example, consider we are searching for the term 'apple'. Entering our query term in a Web search engine, e.g., Google, retrieves a huge list of matching pages (around 341,000,000). The first results are all related to the company 'Apple'. No pages from other categories are shown. But, especially for high recall searches, it would be more helpful to retrieve a better structured result set offering a suitable overview of the general Web page categories.

Actually, there are already approaches that cluster the results and offer a set of general categories for filtering. An example is the search engine Clusty[33]. If we search for the word 'apple' again, we get a number of categories to constrain the result set,

[32] Yahoo Heat maps: Web search engine metrics tutorial at World Wide Web Conference in 2009

[33] http://clusty.com

e.g. 'store', 'reviews' or 'tablet'. However, the problem still is that categories that occur in more pages are considered to be more relevant. Therefore, we do not get a complete overview of the whole category dimensions and are still focused on the company.

To enable navigational browsing in an online portal, a high quality ontology is still beneficial. In chemistry, only a few ontologies are openly available. These ontologies are focused on specific sub-domains, consider, e.g., ChEBI[34]. But, a freely available, community maintained knowledge base is offered by Wikipedia. During the last years, the coverage of Wikipedia has reached a large pool of information including articles from almost all areas. Each article is assigned to a number of categories, which are hierarchically ordered and form a shallow ontology. Recent work proposes to use these categories to identify the topics of a document. But, consumers have different workflows and expectations when searching for literature, depending on their scientific domain and the task that should be solved. Does Wikipedia also provide a valid knowledge base for specific domains like chemistry?

In this section, we analyze whether the Wikipedia categories system is also useful for describing specific domain knowledge. We take a document collection from the open access journal *Archive for Organic Chemistry* (ARKIVOC) and assign Wikipedia categories to each document. Furthermore, we also assign the terms of the domain-specific ChEBI ontology to each document. We then represent each document of the collection by a Wikipedia categories tag cloud and a corresponding ChEBI tag cloud. A survey with a team of domain-experts was used to evaluate the different representations and assess the degree to which each representation is useful.

5.1. Related Work

The goal of topic identification is to find any kind of labels, like, e.g., categories or keywords, describing the content of a document. There are several approaches aiming at detecting document topics using a fixed set of predefined labels. In [89] an approach is presented automatically identifying a topic of a Web document by exploiting the Yahoo! directory. Since the vocabulary of this Web ontology is limited, the concepts are enriched by an external linguistics knowledge base (WordNet). The process of extracting the most important entities is based on HTML tags. They choose sentences or words that occur in title tags or that seems to be important for the document, e.g., which are highlighted or point to other documents. Finally, the resulting terms are mapped to the ontology nodes. They evaluated their approach by computing precision values and comparing them to approaches that use machine learning techniques for document classification. Still, the accuracy of the machine learning approaches is slightly better.

[34] http://www.ebi.ac.uk/chebi

In the last years, the size and coverage of Wikipedia has reached a size where recent work proposes to use it to identify topics of documents using the Wikipedia category network. This category network can be seen as a simple ontology, which is used by many users and constantly refined by Wikipedia editors [90]. An approach quite similar to the one described in [89] is discussed in [91]. Here the titles and categories of Wikipedia articles are used to characterize documents. After stopword removal and stemming, a weight is assigned to each word of the source document. Afterwards, all Wikipedia titles and related articles supported by words in the document are collected and weighted. Finally, the assigned categories are retrieved and ranked. The top-k categories are used to describe the documents' content. The approach was evaluated by predicting categories of Wikipedia articles and discovered topics were subsequently used for clustering. Interestingly, in general scenarios Wikipedia categories seem more useful to describe documents than the respective fulltext.

Other Wikipedia related approaches focus on improving text classification performance by enriching document representations with Wikipedia concepts, see, e.g., [92], [78]. Here, a feature generator, which acts as a retrieval engine, is responsible for the mapping between documents and Wikipedia concepts. This generator uses single words, sentences, paragraphs or the whole document and outputs the most relevant Wikipedia articles. Finally, the titles of the retrieved articles are used as additional features to enrich the document representation. However, the mapping between documents and Wikipedia concepts relies on an exact phrase matching strategy. Therefore, the coverage of topical terms from the documents and related Wikipedia articles is limited. If two documents use topical terms, which are not directly matched to a Wikipedia article but are synonymous, these articles will not be assigned to the documents. We also use an exact match strategy in our approach, but since our search is based on chemical entities, we already include all synonyms in the enriched index pages, respectively our database. Furthermore, these methods only consider Wikipedia concepts and do not consider the hierarchically relationships available in Wikipedia in its categories hierarchy.

Finally, in [80] a clustering method is introduced that uses Wikipedia concept and category information for document clustering. Beside an exact match strategy, also a relatedness-match is presented avoiding the further mentioned synonym problem by not merely using Wikipedia article titles for matching, but also considering the content of the whole Wikipedia articles. The outcome of all these approaches is that Wikipedia is indeed useful for describing and summarizing the content of documents. However, all approaches were focused on general documents, respectively Web retrieval. Considering more specific domains the requirements may differ.

5.2. Generating Tag Cloud Representations

Both knowledge bases, Wikipedia and ChEBI, are freely available as database dump downloads. Whereas the ChEBI tables are directly usable, the Wikipedia database

dump requires two preprocessing steps for data cleaning. The Wikipedia category corpus includes categories that are completely useless for topic identification. Namely categories, like, e.g., *All articles with unsourced statements* or *Wikipedia semi-protected pages* are discarded. Furthermore, the Wikipedia dump includes different types of pages, which are marked with different namespaces. Beside the 'normal' page namespace, Wikipedia currently has 21 additional namespaces, like, e.g., User, Template, or Category. Since we are only interested in main pages, we only used pages with namespace 0 and discarded other possible hits from the result set.

For each document, we have a list of all extracted chemical terms. Each term is mapped to a Wikipedia page and a ChEBI ontology node. Since we are not interested in the Wikipedia pages themselves, we only extracted the associated category entries. Starting from this entry point all parent categories are extracted and appended to the chemical term. We did the same for the ChEBI ontology nodes. Each chemical term is described by a set of categories and a set of ontology nodes. Hence, the documents are described as the union of the category/ontology node sets of all included chemical entities. Finally, each document is represented by two different tag clouds, one containing the ChEBI nodes and the other containing Wikipedia categories. The cloud terms are weighted by their frequency. The following algorithm summarizes the different steps:

For each **document** in the collection do
1. Get a list of all included entities
2. For each **entity** in the list do
   2.1. If (ChEBI)
      2.1.1. Find matching ontology node in ChEBI
      2.1.2. Compute path from that node to the root and add the complete path to document's ChEBI vector
   2.2. Else
      2.2.1. Find Wikipedia page for that entity
      2.2.2. Get all categories and their parents up to the root for that page and add them to the document's category vector.
3. Compute Category / ChEBI cloud

Algorithm 5. Creating tag clouds

Again, as document repository we use 2700 chemical documents from the journal *Archive for Organic Chemistry (ARKIVOC)* and the corresponding enriched index pages.

Our evaluation contains four different experiments:

- Since document retrieval in the chemistry domain is centered on chemical entities, we evaluated in the first experiment where to find the most important chemical terms regarding the document context. Therefore, a group of domain experts manually evaluated the importance of each chemical term occurring in a document.

- In the second experiment, we took the important terms found in our document collection and retrieved all associated Wikipedia categories. A team of chemists evaluated if these categories are useful for describing chemical documents.
- In the next step, we want to know whether the mapping of the chemical terms to ChEBI ontology nodes, respectively Wikipedia categories, is comprehensible. We analyzed the distribution of the important terms from our document collection and choose a random subset. For each term in this subset, we retrieved the matching Wikipedia categories and ChEBI ontology nodes. Again, a team of domain experts evaluated the mapping quality of the associated nodes/categories. Furthermore, we evaluated this quality based on the level in the category/ChEBI tree.
- In the last experiment, we created tag clouds based on the category information and the ChEBI ontology nodes associated with each document. Our group of domain experts evaluated the quality of these clouds by stating how the content of the document is represented by the terms in the clouds.

5.2.1. Where To Find The Most Important Entities?

Usually a huge number of chemical entities are mentioned in domain-specific documents. However, not all of them are relevant for describing the document's content. Especially the term frequency is not a useful measure for chemical terms, since frequent occurring terms in most cases are actually not important, e.g., solvents like *Benzene* or catalysts like *Palladium*.

Our first experiment identified the parts of the document where the most descriptive chemical terms occur. Therefore, we took a random set of documents from our collection and their enriched index pages. We then delivered the pages to a team of domain experts who marked the most descriptive entities for each document. We observed that in most documents the relevant terms are already mentioned in the title and/or the abstract. Nevertheless, in some abstracts placeholders are used to link to a complex structure drawn in an image or to other complex textfragments. For example, the 3 in *hexabromide 3* is linked to an image visualizing the complex structure of: *1,2,3,4,9,10-hexabromo-1,2,3,4-tetrahydroanthracene*. As mentioned earlier, especially the information contained in drawn representations is currently not automatically extractable.

Many documents also include an experimental part where the different synthesis steps are shown. In this part, a lot of abbreviations for entity names are used leading to many entity fragments extracted by the entity recognition module. **Fig. 50** shows an extract of a document's experimental part.

Hexabromide **3** (2.5 g, 3.8 mmol) was dissolved in dry and freshly distilled pyridine (50 mL) at 0 °C. The reaction mixture was stirred at room temperature overnight. After completion of the reaction (TLC control), pyridine was removed *in vacuo*. The residue was diluted with ether (50 mL) and washed with HCl solution (1.56 M, 100 mL). After removal of the solvent, the precipitated material (tribromide **12**) was filtered through a short silica gel column (10 g, eluting hexane) and recrystallized from chloroform/hexane (1.19 g, 75%), yellow needles, m.p. 169 °C . ^{1}H NMR (400 MHz, $CDCl_3$) δ 8.75 (s, 1H, H 1), 8.55 (m, 2H, H 5 and H 8), 8.3 (d, 1H, J_{34} = 7.6 Hz, H 3), 7.57 (d, 1H, H 4), 7.57 (m, 2H, H 6 and H 7); ^{13}C NMR (100 MHz, $CDCl_3$) δ 133.1, 133.0, 132.8, 132.1, 131.9, 130.3, 130.2, 130.0, 129.7, 129.4, 125.7, 125.5, 124.4, 124.1; IR (KBr) ν_{max} 3025, 1616, 1602, 1438, 1421, 1292, 1072, 1065, 862, 802, 748; MS (CI) *m/z* 411.77/413.76/414.77/415.76/417.76 (M^+), 331.88/333.88/334.88 /335.88/336.88 (M^+-Br), 254.00/255.00/256.00/257.00 (M^+-2Br), 173.08/174.08/175.08/176.08 (M^+-3Br), 149.08/128.05/127.05/110.06; Anal. calcd for $C_{14}H_7Br_3$ (414.9): C, 40.53; H, 1.70. Found: C, 40.65; H, 1.63.

Fig. 50. Annotated part of the experimental section

The highlighted text fragments were annotated by the recognition module. While some of them, like *pyridine* or *hexane*, are correctly annotated, the majority are only invalid text-fragments, like, e.g., *C* or *H*. To reduce the number of invalid text-fragments we only consider terms from title and abstract for the following evaluations.

5.2.2. Wikipedia Category Suitability

Especially for a specialized field like chemistry the coverage of Wikipedia for chemical terms seems questionable. We took all chemical terms found in our document collection (title and abstract) and tried to find suitable Wikipedia pages. In total, our collection contains 11,952 distinct chemical terms. Surprisingly, for 2163 we found an entry page in Wikipedia (18%) which include 745 distinct categories in total. Now, a team of domain experts analyzed the set of Wikipedia categories found for those pages. For each category, our experts rated whether it is useful for classifying chemical documents or not. Each category is assigned with a value ranging from 0 (not relevant) to 4 (very useful). More than 25% are rated as good, but only two categories were considered very useful for classifying chemical documents: *addition reactions* and a*minoglycoside antibiotics*. However, more than 50% are at least not bad and can in principle be used for document classifying.

5.2.3. Mapping Traceability

During this evaluation, we analyzed the traceability of the mapping of chemical terms to ChEBI ontology nodes, respectively Wikipedia categories. To find a set of suitable query terms, we need to know the distribution of the chemical terms in our open access journal collection (see **Fig. 51**).

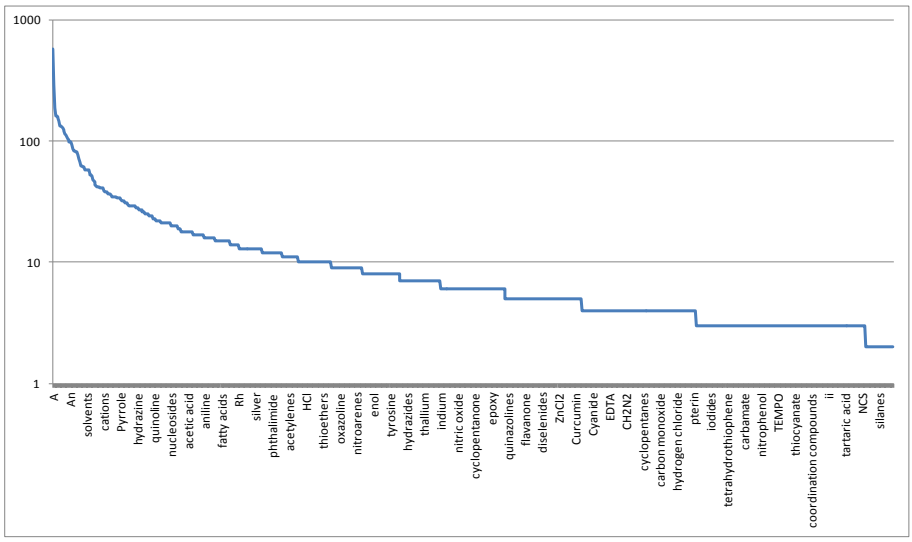


Fig. 51. Chemical term distribution

In total, we found entries for 2163 disjunct terms occurring in title and abstract of the documents from our collection. Due to the automatic extraction process there are some recognition failures, e.g., the most often occurring term is *A* with 579 hits. For the query evaluation, we therefore choose a representative subset. We took all terms occurring between 20 and 100 times resulting in a set of 129 chemical terms. Terms that occur more often are too general or can be lead back on recognition errors. Entities occurring less than 20 times in the whole set are too specific and have little chance of finding appropriate matches in Wikipedia. From this subset, we choose approximately 10% as query terms (resulting in 12 queries). For each query term, we retrieved the matching ChEBI ontology node and all their parent nodes up to the ontology's root node. 121 of these terms were found in the ChEBI ontology (94%). Furthermore, we searched for the matching Wikipedia page and extracted their categories. Here, 79 of the 129 terms have a matching Wikipedia page (61%). Every Wikipedia page is assigned by at least one category. For each category, we took all parent categories up to the root node. Finally, we have for each query term a set of ontology terms and a set of Wikipedia categories. These sets are evaluated by a group of domain experts who rated for each category how close it is related to the query term. The value range is from zero (not relevant) to five (very relevant). **Table 15** shows the average values for each query term.

Table 15: Scores for sample queries

Term	ChEBI Score	Categories Score
Rhodium	1,8	2,4
Pyridine	0,7	1,8
Ester	0,4	0,5

Term	ChEBI Score	Categories Score
Phenol	0,4	1,4
Alcohol	0,4	0,4
Oxygen	1,2	0,7
Sulfur	1,6	0,8
Pyrazole	0,8	1,9
Zinc	1,1	1,0
Aldehyde	0,3	1,3
Copper	1,6	0,8
Palladium	2,6	1,8

The average scores are quite low for all queries. The problem is that we considered too many levels in the category hierarchy as well as in the ChEBI ontology. The more general the category, respectively the ontology node, the lower is the associated score. **Fig. 52** visualizes the score distribution based on the level. Here, the level means the number of edges that needs to be passed to reach a node starting at the query node. For example, level one includes all terms directly linking to the query term. Please note that for the Wikipedia categories each query term can retrieve more than one category node. Each of these categories is a leaf node in the combined categories tree.

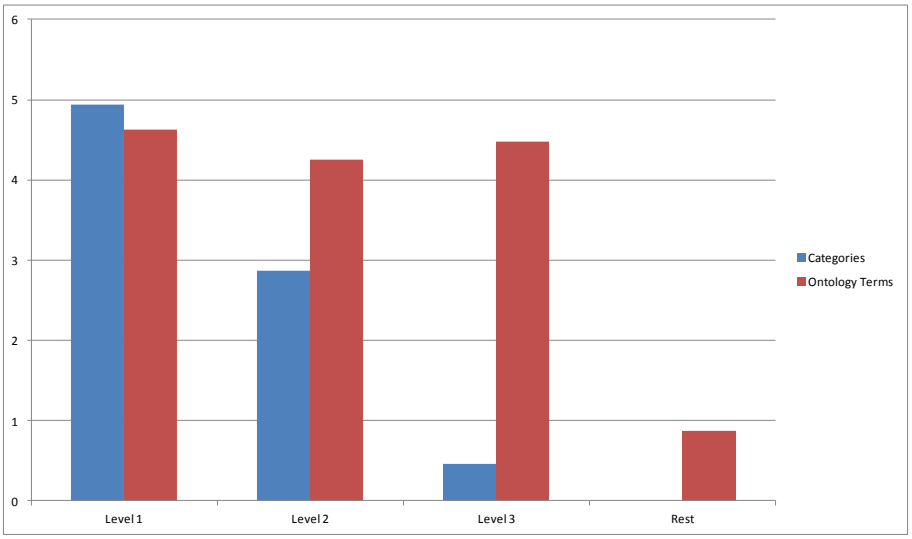


Fig. 52. Level-based score distribution

For the domain-specific ontology, the first three levels are almost equally important, whereas for the more general Wikipedia categories only terms of the first level are really relevant. The reason is that Wikipedia includes knowledge from many

different domains. If we explore the categories graph, we see that only a few steps away from the query term, we reach categories that are not relevant for our domain. For example, **Fig. 53** shows the level 2 categories graph for *Palladium*. Only two nodes away from the query term we reach the nodes *Currency* or *Wealth*, which are definitely not relevant regarding the domain of chemistry. We can also see categories like *Atoms* or *Matter*, that are already far too broad for sensibly categorizing the scope of a journal paper.

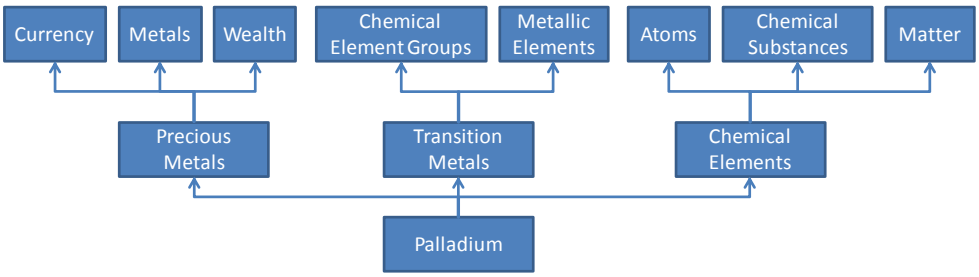


Fig. 53. Level 2 category graph for Palladium

Obviously the more level one goes up in the categories graph, the more general the information gets. Therefore, to avoid unrelated categories we only considered Wikipedia categories that are directly linked to the query term (level one). **Fig. 54** shows the ChEBI ontology graph for *Palladium* that only consists of chemical terms.

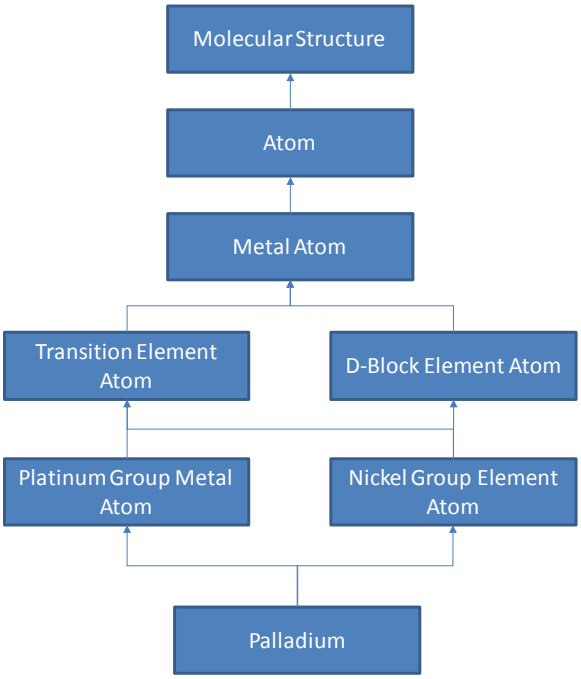


Fig. 54. ChEBI ontology graph for Palladium

Please note that this graph includes all nodes from the query node up to the root node. The fourth level also already contains one element, which is very general,

namely 'Atom'. Since our evaluation has shown that this is a general observation, we only considered nodes that are at most three levels away from the query node for the ChEBI ontology.

Comparing the categories and the ChEBI graph, it is interesting that the different levels in the ChEBI graph include fewer nodes. **Table 16** gives an overview of the number of associated ontology/category terms for chemical entities. The values are averages and refer to the representative subset of the 129 chemical terms. Using the level information, the number of associated terms has been highly decreased.

Table 16. Average number of associated terms for each chemical entity

	With Level Restriction	**Without Level Restriction**
Wikipedia	4	76
ChEBI	11	39

5.2.4. Comparing Wikipedia Categories and ChEBI Ontology Terms

In this experiment, we analyze the coverage of ChEBI ontology terms and Wikipedia categories. How many ontology terms are included in Wikipedia? In total, we evaluated 16056 ontology terms and searched for corresponding Wikipedia entries. Only 757 of them have a Wikipedia entry (4.7%). While also relatively complex chemical substances, e.g., *5-Methoxy-N,N-dimethyltryptamine*, have suitable Wikipedia pages, ChEBI contains a lot of even more complex substances, e.g., *4,5,9,10,14,15,19,20,24,25-decaethyl-26,27,28,29,30-pentaoxahexacyclo [21.2.1.1(3,6).18,11.1(13,16).1(18,21)]triaconta-1(25),3,5,8,10,13,15,18,20,23-decaene*. Moreover, also 3642 InChI-codes are included, which have no Wikipedia entries. The minimal overlap of the two knowledge bases of only 4.6% leads us to another experiment to evaluate which knowledge base is more useful for describing chemical documents. We randomly took a set of documents from our collection, extracted all chemical terms and searched for matching ChEBI terms and Wikipedia categories. Afterwards, we build tag clouds; one for the ChEBI terms and another for the Wikipedia categories. Each document is represented by its title, abstract and both clouds. Our team of domain experts rated how good each representation summarizes the document's content. The results are shown in **Table 17**. Here, the rating values range is from zero (not relevant) to five (very relevant).

Table 17: Average scores for first weighting scheme

Title	3.2
Abstract	4.5
Categories Cloud	1.75
ChEBI Cloud	0.65

The abstract gives always a good overview of the document's content. The title is still representative, but the other representations are far behind and not usable off-hand. It is interesting that the categories cloud is more useful than the clouds generated from the domain-specific ChEBI ontology. Please note that the clouds only include the top-30 terms ranked by their frequency. The problem of this first approach is that we considered the term frequency for computing the respective clouds. As mentioned earlier, for the area of chemistry the term frequency is not the preferred weighting scheme. In most cases, the frequent occurring terms (which are often solvents, catalysts, etc.) have no impact on the actual topic of the document. The really descriptive terms are occurring in the title or abstract and are usually rare. This leads to relatively unspecific clouds and therefore, to low rating values. **Fig. 55** shows an example of a categories cloud.

Fig. 55. Example: Wikipedia category cloud

This cloud includes relatively general concepts, like, e.g., *chemical elements*, as well as categories that are not related to the domain of chemistry, like, e.g., *airship technology* or *spoken articles*. The average score assigned from our experts team is 1, meaning that the cloud does not really describe the content of the document. **Fig. 56** shows the ChEBI ontology cloud for the same document. It includes totally different terms. The reason is, as we saw before, that the knowledge bases only have a minimal overlap.

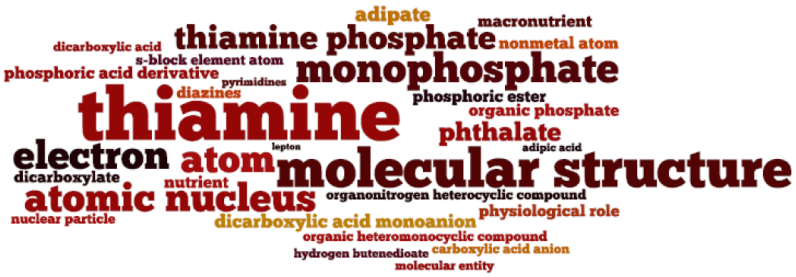


Fig. 56. Example: ChEBI ontology cloud

The average score of this cloud is 0.5. The problem is that the mentioned terms are too general. Terms like, e.g., *molecular structure* or *atomic nucleus*, are not very descriptive for a document. This results in too general, not descriptive clouds.

In a second experiment, we changed the weighting scheme for the terms by weighting seldom occurring terms higher. **Fig. 57** shows the Wikipedia categories cloud for the same document using the other weighting scheme.

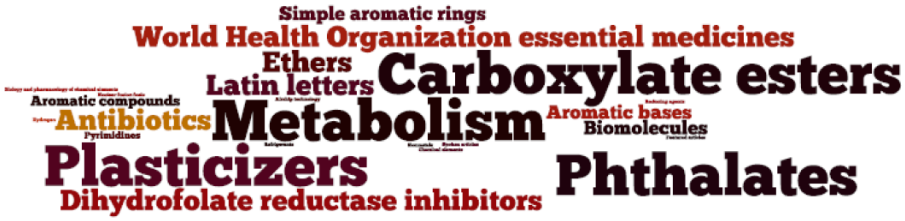


Fig. 57. Example: Wikipedia category cloud, different weighting scheme

This cloud got an average score of 4, meaning to offer a good description of the corresponding document. The terms occurring in this cloud are more specific, like, e.g., *carboxylate esters*. Of course, some terms are also not useful, like, for instance, *latin letters*, but nevertheless, most terms give a good overview of the documents topic. **Fig. 58** shows the ChEBI ontology cloud for the same document.

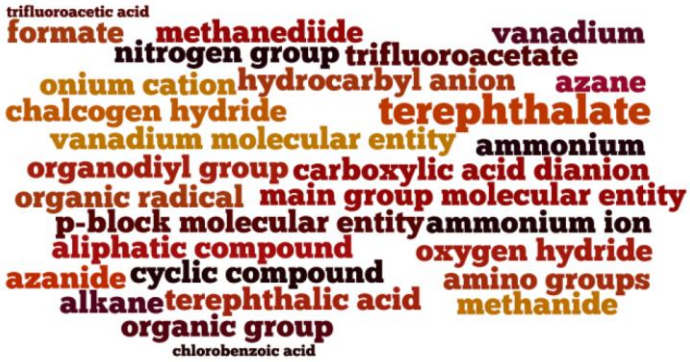


Fig. 58. Example: ChEBI ontology cloud, different weighting scheme

We see, that compared to the other ChEBI cloud the terms have completely changed being a lot more specific. Although the average score for this cloud also increases (1), to the means of our domain experts, it still does not describe the document's content as well as the categories cloud. **Table 18** shows the average scores for this second weighting scheme:

Table 18: Average scores for second weighting scheme

Title	3.2
Abstract	4.5
Wikipedia Categories Cloud	2.8
ChEBI Cloud	2.2

Since we used the same documents, the title and abstract scores have not changed. However, the categories and ChEBI scores have slightly increased. To conclude, we want to state that it is really surprising that from the domain experts view the Wikipedia clouds are more descriptive for chemical documents than a domain-specific knowledge base, like the ChEBI ontology. We already mentioned that the domain of chemistry includes many different working areas. The chemists have a different understanding of the relations between chemical substances heavily depending on the area they are working in. We did a final survey with a team of chemists to analyze if the relations between entities in the ChEBI-ontology are comprehensible. The results are that even for chemists from the area of organic chemistry not all relations in the ontology are comprehensible. Thus, for them, the mapping of the ontology terms to the chemical entities is also invalid. Indeed, the Wikipedia categories offer a suitable alternative to domain-specific ontologies.

5.3. Conclusions

To give the user a good overview of the documents' content in his/her result set, we presented an approach using Wikipedia categories to generate compact representations of chemical documents. As a baseline, we used a domain-specific ontology (ChEBI) to represent the documents and compared the results. Each document from our repository is described by a Wikipedia categories cloud and a ChEBI ontology cloud. Our evaluation by a team of domain experts has shown that the Wikipedia categories are even more expressive for describing chemical documents than the handcrafted, domain-specific ChEBI ontology terms. Therefore, we have shown that the Wikipedia categories system can be used in domain-specific portals to overcome the problem of expensive, manually created ontology knowledge.

Chapter 6

An Architecture for Chemical Digital Libraries

In the last chapters, we presented different steps enabling semantically enriched text-based retrieval in chemistry. First, we created enriched index pages building the basis for text-based retrieval. Since users are interested in finding similar entities regarding their query, we further analyzed different similarity measures in chemistry. We found out that users have specific background knowledge influencing their subjective notion of relevance. To model this implicit knowledge, we presented an approach clustering chemical entities based on their functional groups. Moreover, users are often interested in chemical entities occurring in certain contexts within the documents. This contextual information is important to assure high quality retrieval results. The presented approaches use external knowledge bases and cross-domain ontology knowledge to enable contextual queries in chemistry. We also showed an alternative representation of the retrieval results to give the user a good first impression about the content of the retrieved documents. Again, for the presentation of the documents, we used external knowledge to enrich the documents with suitable metadata. In this chapter, we combine these different steps and build an architecture for a chemical digital library. Since for almost all steps different information sources are required to allow for high quality retrieval, we show how to integrate them in the workflow of a digital library. Of course, most of the semantic enrichments can already be preprocessed to save computation time during retrieval. In addition to textual queries, our architecture also contains components enabling chemical structure queries. We will briefly explain what is needed during preprocessing and how the retrieval using a graphical interface works.

6.1. Preprocessing: Index Page Generation and Semantic Metadata Enrichment

In this section, we describe which metadata can be generated in a preprocessing step. Anyway, the documents have to be indexed and all chemical entities have to be extracted. But, beside these mandatory steps also some additional steps can be performed to create semantic metadata that is needed to provide high quality retrieval. These steps can easily be integrated in the indexing workflow of a digital library. **Fig. 59** gives an overview of the whole preprocessing workflow. First enriched index pages are generated, which are further extended with semantic metadata. We explain both steps in the following sub-chapters.

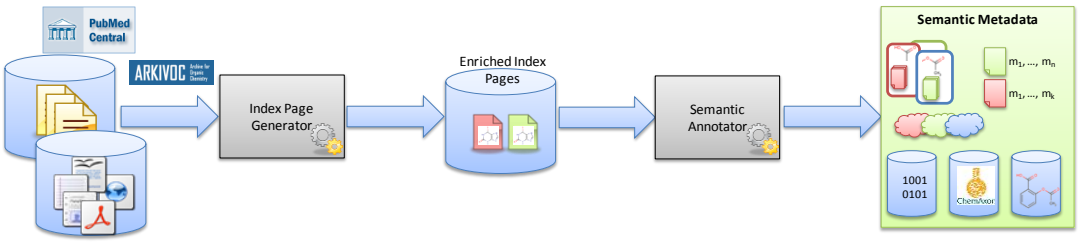


Fig. 59. Preprocessing workflow

6.1.1. Creating Enriched Index Pages

We already gave a detailed description of the necessary steps to enable text-based retrieval in Chapter 2. An overview of these steps is shown in **Fig. 60**.

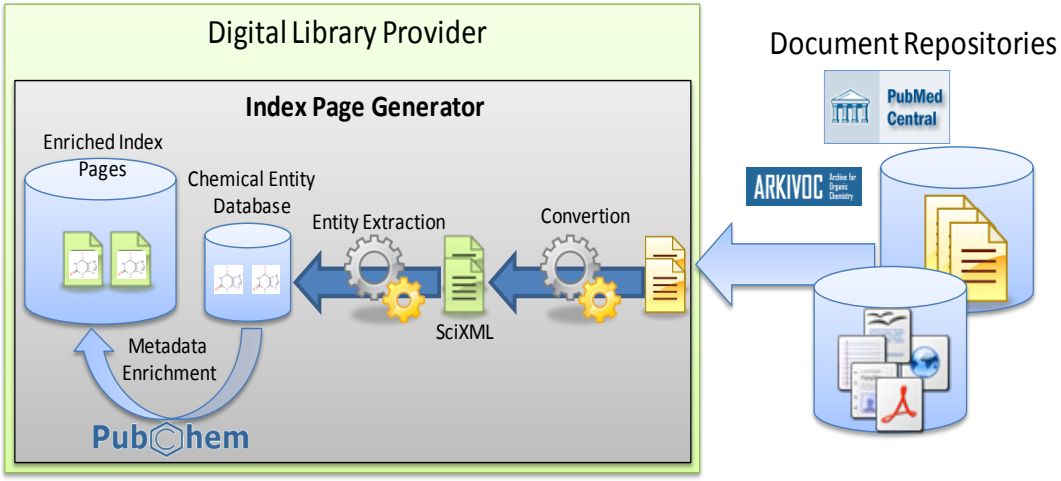


Fig. 60. Index Page Generator

The first step is to convert the different documents into a uniform format, which in our case is SciXML. The SciXML documents are used by OSCAR, which annotates all included chemical entities. Afterwards, the extracted entities are enriched by synonyms and other representations, like, e.g., SMILES or InChI. For the enrichment with suitable metadata, we use the domain-specific PubChem database. PubChem contains facts about around 120 million chemical substances. For each document, this information is combined in an enriched index page, which is stored in the repository of the digital library provider and used for text-based retrieval. In addition, the chemical entities are stored in a separate database and are linked to the documents, respectively the enriched index pages, they occur in.

6.1.2. Semantic Metadata Enrichment

Starting from the enriched index pages a lot of valuable metadata can be created during preprocessing. We create several components, each of them responsible for

creating specific metadata. These components are combined in the semantic annotator. **Fig. 61** gives an overview of the different components used in the semantic annotator.

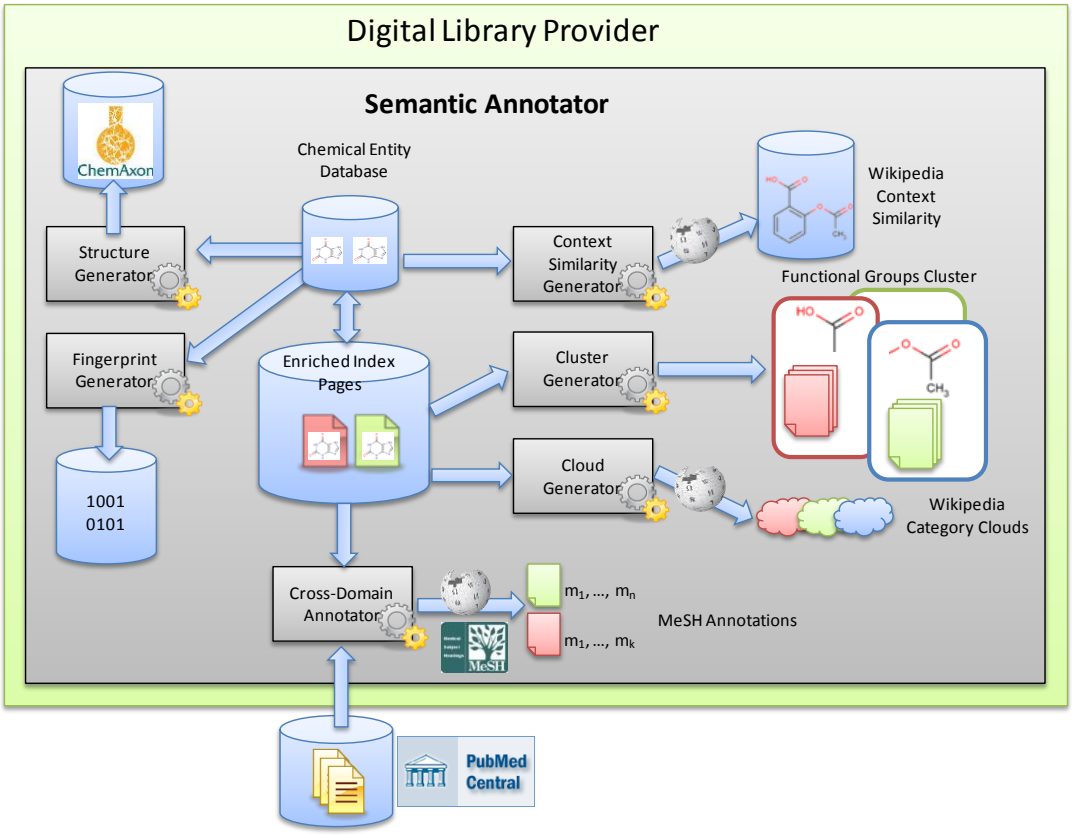


Fig. 61. Semantic Annotator

Cluster Generator: We saw in Chapter 3.2 how to reflect the chemist's perception of chemical entities belonging to the same chemical class. We modeled this implicit knowledge by clustering chemical entities based on their functional groups. Each cluster describes a class of entities with similar reaction characteristics. The first step is the generation of the functional groups cluster. Therefore, all chemical entities from the database are taken and the functional groups of each entity are extracted using the extended checkmol tool (see Chapter 3.2.1). Each entity is associated to the respective cluster based on its functional groups. Afterwards, the clusters containing more than 100 chemical entities are further decomposed by computing sub-clusters based on the substructure fingerprint and the Manhattan distance. In the second step, the enriched index pages are associated to the respective clusters based on their contained chemical entities. Hence, each document can be associated to several clusters.

Context Similarity Generator: As presented in Chapter 4.2, the context similarity scores are based on Wikipedia and can be precomputed. Each term having its

own Wikipedia page can be used as context term. Therefore, it is possible to precompute the context similarity of each term in the Wikipedia repository to each chemical entity. Alternatively, as useful subset of basic context terms the Wikipedia categories can be used. The similarity values are computed using the relatedness measure and are stored in the context similarity database.

Cross-Domain Annotator: The insights of this component are explained in Chapter 4.3. It is responsible for annotating all chemical documents with suitable cross-domain ontology terms. As cross-domain ontology, we use the MeSH ontology from the related domain of biomedicine. All documents from the PubMed Central repository (PMC) are annotated with several terms from the MeSH ontology. Since these annotations are done manually by domain experts, they are of high quality. We extracted all chemical entities from the PMC documents and used them as features to learn the MeSH terms. For each MeSH term, a classification model is learned based on its associated chemical entities. These models are further used to automatically annotate chemical documents with MeSH terms. To increase the quality of the associated terms Wikipedia is used as a semantic filter. All terms that are not semantically related to the documents are removed. Finally, we store the associated MeSH terms for each chemical document.

Cloud Generator: To give the user an idea of the documents' content, we create compact document descriptions using Wikipedia (see Chapter 5.2). For each chemical entity from our entity database, the corresponding Wikipedia page is retrieved. The chemical entity is described by the set of associated Wikipedia categories. As shown in the experiments in Chapter 5.2 only directly associated categories are used. The documents are described as the union of the Wikipedia categories of all contained chemical entities. Finally, each document is represented by a tag cloud containing Wikipedia categories weighted by their inverse frequency.

Fingerprint Generator: The basis of all structure-based similarity measures is a fingerprint representation of chemical entities (see Chapter 3.1). Instead of computing these fingerprints during retrieval, for each chemical entity in the entity database the corresponding fingerprint representations are precomputed and stored in a database. To compute the fingerprints, we rely on the Chemical Development Toolkit (CDK), which supports six different chemical fingerprints [33], [34]. To generate the fingerprints a structural representation of the chemical entity, like, for example, SMILES, is needed. This representation is available for almost all entities in the repository. If it was not automatically created by OSCAR's name-to-structure algorithm during the entity extraction, it was added during the metadata enrichment process in the index page generator.

Structure Generator: As stated in the introduction (see Chapter 1) chemists also need the possibility to search for chemical structures. To allow for graphical

queries a special chemical structure databases is used. We rely on the JChem framework provided by ChemAxon[35]. Of course, most of the factual data and metadata could also easily be stored in some arbitrary relational database system like MySQL. However, for efficiency and improved handling the structural data is stored in the specialized chemical structure database JChem Base.

6.2. Semantically Enriched Retrieval

In this section, we combine the different retrieval workflows and present an architecture for personalized retrieval in chemical digital libraries. **Fig. 62** gives an overview of the proposed architecture.

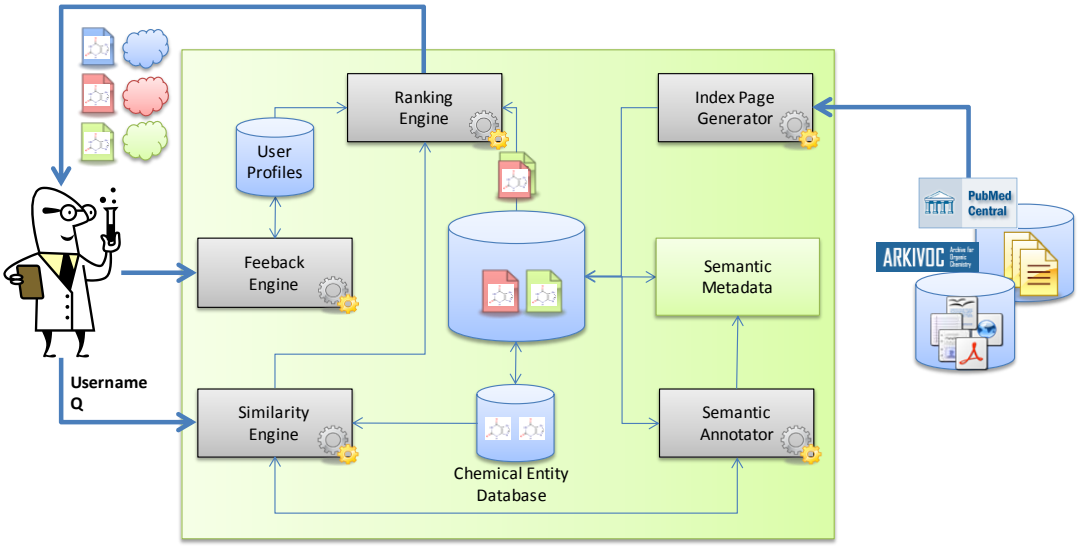


Fig. 62. Personalized retrieval architecture

The user submits a query *Q* and his/her username to the system. Dependent on the query type the similarity engine has to decide which steps are necessary to find a suitable result set. The username is needed by the ranking engine to get the user profile. The user profile includes information about the search history, like, e.g., the preferred similarity measure.

Structure query: In case the user submits a query using the graphical query interface, the similarity engine uses the ChemAxon framework to process the query. Chemists differentiate between exact structure searches, substructure searches, and similarity searches. While the exact search will return only exact matches, the substructure search will return all structures including a given substructure. A similarity search will return structures based on the calculation of a similarity match, which can vary from database to database. In any case, to search for matching structures a rapid prefiltering step is performed. In this step, many of the targets not matching

[35] http://www.chemaxon.com

the query are screened out. It is based on chemical hashed fingerprints. Since it may result in false hits, the results are further checked using a precise atom-by-atom search. This check increases the retrieval quality, but makes the search much slower. Of course, these steps can be manually tuned if necessary. The set of matching entities is handed on to the ranking engine, which retrieves the related documents and delivers the associated Wikipedia tag clouds to the user.

Text-based query: The query is composed of the chemical entity q_e and the context term q_c. To make it easier for the system to distinguish the query terms we introduce a simple colon notation: *context:* for q_c and *entity:* for q_e. If q_c is null a basic similarity search is performed.

Similarity search: Since our enriched index pages also contain structural information for the chemical entities (SMILES), it is possible to use the structure-based similarity measures for textual queries. All these measures are based on fingerprint representations of the chemical entities. Since these fingerprints have already been precomputed and stored, computing the similarities is much faster than for structure queries posed via a graphical query interface. Moreover, the set of available entities is further constrained to those entities that are in the same functional groups cluster as the query entity. To find the cluster, the similarity engine interacts with the semantic annotator to get the functional groups cluster of q_e. If q_e is not known that means that there are also no matching documents in the system. All chemical entities and their synonyms have been associated to their functional groups cluster during preprocessing. Otherwise, the similarity engine receives all chemical entities from the respective cluster and hands the set on to the ranking engine. The ranking engine computes the fingerprint-based similarities of q_e to all chemical entities in the functional groups set. To know which measure to use it analyzes the user profile. In case the user is new to the system, the globally most preferred measure is used. This choice is later adapted using user feedback. Afterwards, the enriched index pages containing the chemical entities are retrieved. The set is ranked based on the entity similarities.

Contextual search: In case that the context term is not null, the first step is to check whether it is contained in the MeSH ontology. If yes, the similarity engine tells the ranking engine to use the cross-domain terms for context filtering. If not, the context term is handed on to the semantic annotator. If the context term is known to our Wikipedia context similarity database, the similarity engine tells the ranking engine to use Wikipedia context similarity for ranking the documents. Otherwise, the semantic annotator computes the context similarity of q_c to all other chemical entities known in our system. In case q_c is not known by Wikipedia, we cannot consider it as valid context term. In this case, only a fulltext filtering is possible, meaning only documents are returned containing q_c in the fulltext.

For both cases, similarity and contextual searches, the documents are ranked based on the most preferred similarity measure for the respective user. This measure is found in the user profile. When a context search using Wikipedia context

similarity is performed, the most suitable feature combination for the user is used for entity similarity (see Chapter 4.2.1). Otherwise, the most preferred fingerprint-based similarity measure is chosen. Instead of showing the documents directly to the user, the result set contains the associated Wikipedia tag clouds of the documents. Thus, the user gets a first impression of the documents content. If the document seems interesting, the fulltext can be directly accessed using a link on the index page.

The user has the possibility to give feedback to our system by marking retrieved documents as irrelevant. All documents, which are not marked as irrelevant are treated as relevant hits. The judgments are collected by the feedback engine. The collected feedbacks are used by the ranking engine as a gold standard to compute all possible rankings using the uncorrelated similarity measures. Finally, the user's profile information is updated if the best matching similarity measure changes. Because the computation may be time consuming, it is not done during query time. The user profile is updated if the user logs out of the system. The adapted profile is available after the computation has finished.

6.3. Conclusions

In this chapter, we combined the findings from the previous chapters and presented an architecture for a chemical digital library. The most important part to enable high quality retrieval is the metadata enrichment process. Metadata enrichment plays a major role for almost all parts we presented:

- For creating the index pages knowledge provided by the PubChem database is needed to enrich the chemical entities.
- To enable similarity computations all chemical entities are converted into several different fingerprint representations.
- For context annotations, MeSH ontology terms and Wikipedia knowledge are used to enrich chemical documents.
- To model the chemists' implicit knowledge, the chemical entities of the whole collection are clustered based on their functional groups.
- To give the user a good overview of the documents' content, Wikipedia categories are used to create meaningful tag cloud representations for the documents.

The digital library provider can precompute all required metadata already in the preprocessing phase to save computation time during retrieval. The whole metadata generation can easily be integrated in the indexing process, which is mandatory anyway.

In the second part of this chapter, we presented a retrieval workflow supporting the different types of queries a user is interested in. The proposed architecture is composed of several components necessary to enable high quality retrieval. By using a feedback engine, we further increased the retrieval quality by learning the preferred similarity measure for each user.

Chapter 7

Conclusions and Future Work

Today, the access to chemical information is often based on complex structure searches requiring specialized indexes and graphical query interfaces. Nowadays, the most prominent provider of chemical information is the Chemical Abstract Service (CAS), providing a manually maintained, high quality digital library. Of course, the access is quite expensive and strictly limited to subscribers. Obviously, for the growing open access movement expensive, manually maintained structure indices are no suitable alternative. But, nevertheless, it is important to open up their knowledge to practitioners in the chemical domain. There are also freely available search platforms, like ChemXSeer or ChemSpider, offering access to chemical literature. Most of them offer interfaces for textual queries. However, the text-based search capabilities are still on a very basic level. The need for more sophisticated search capabilities was already discovered by several groups. For example, the ChemXSeer platform provides a specialized search index based on chemical formulae. Although the general need for more suitable text-based retrieval methodologies was already discovered, there is still a lot of room for improvements.

Therefore, in this thesis we introduced different steps necessary to enable semantically enriched text-based retrieval in the chemical domain. We started with the creation of enriched index pages to enable basic text-based retrieval. The proposed approach collected different entity representations and synonyms. For each document, an enriched index page has been created and our experiments showed that their retrieval quality is almost as good as for chemical structure searches.

Since users are usually interested in chemical entities not exactly matching the query entity, but sharing similar properties, we analyzed different fingerprint-based similarity measures. There are many different measures available in chemistry and our evaluations showed that many of them are also uncorrelated. We tried to assign them to specific search tasks, but that was not possible. One possible solution is to learn the best measure for each user in a personalized retrieval system. We discussed with domain experts why so many uncorrelated measures are available in chemistry. We figured out that the reasons are that each chemist has specific background knowledge in mind influencing his/her perception of relevance. This background knowledge can hardly be expressed in a query. We figured out that this knowledge is based on the chemist's implicit knowledge about chemical classes. These classes can be described by grouping together chemical entities showing the same or similar reaction behavior. We were able to model chemical classes by clustering entities based on their functional groups. Our evaluations have shown that

the clusters are of high quality. Using the clusters we reduced the amount of chemical entities that needs to be considered for retrieval by around 90% without losing relevant information. Our experiments proved that the proposed approach reflects the implicit knowledge of chemical classes to a large degree.

Now we were able to search for chemical entities using textual queries and also to find similar entities. However, users often search for chemical entities occurring in a specific context. It is very important to also consider this context in the query to allow for high quality retrieval. We showed that structural, fingerprint-based measures are not useful for contextual queries. Therefore, we presented two approaches allowing for contextual searches. Both use knowledge provided by Wikipedia. The first creates profiles of chemical entities by combining different features gathered from the Wikipedia page of the respective entity. The experiments showed that contextual searches are possible using the provided similarity measure. Again, the retrieval results could be further improved using personalization based on user feedback. In the second approach, we annotated chemical documents with cross-domain ontology terms. We learned classification models from the biomedical domain by extracting chemical entities from MeSH annotated MEDLINE documents. Using the learned classifications, we annotated chemical documents with MeSH terms based on their contained chemical entities. To improve the quality of the assigned MeSH terms we used Wikipedia to remove semantically unrelated terms. The experiments showed a strong increase of the retrieval quality compared to baseline retrieval approaches. We further showed the generalizability of our approach by annotating documents from the domain of computer science with cross-domain ontology terms from the related domain of mathematics.

Furthermore, we also presented an approach to present the search results to the user. Since usually a lot of results are retrieved, it is important to give the user a fast and good overview of these results. The proposed approach summarized the documents' content using tag clouds. We compared clouds based on Wikipedia categories to clouds created using the ChEBI ontology. Surprisingly, the Wikipedia clouds have been voted better to describe chemical documents as the domain specific ChEBI ontology clouds. This again proved the usefulness of Wikipedia to semantically enrich the retrieval process also for such specific domains as chemistry.

In the last chapter, we explained in detail which metadata could be extracted and indexed in the preprocessing phase of a digital library provider. Finally, we combined all findings from the previous chapters and presented an architecture for a chemical digital library enabling semantically enriched text-based retrieval.

However, there are still some points, which we leave open for future work. One of the most important parts in the retrieval workflow is the extraction of the chemical entities. If the chemical entities are not correctly extracted, all following steps, like, e.g., semantic annotations are error prone. Therefore, it is important to further improve the quality of automatic entity extraction. In addition, several groups are working on improvements to automatically extract chemical entities drawn in images

within the documents. But, to allow for a fully automatic extraction the detection quality has to be further improved.

For the process of semantic enrichment, it might also be interesting to consider different knowledge bases. In most cases, we focused on Wikipedia. Although the quality of the annotations with Wikipedia is good, it is maybe possible to improve it by combining the knowledge of different information sources. The same applies for cross-domain annotations. It might be very interesting to build a general framework consisting of basic components that can easily be replaced: the document collection to be annotated with ontology terms, the document collection that is already annotated, the used cross-domain ontology, and the knowledge base interacting as the semantic filter. To build such a framework one has to define suitable interfaces on the protocol layer.

Appendix A

Appendix: Role Detection Patterns

Taken Role	Lexico-syntactic pattern in pseudo code
PRODUCT	(?i).* Synthesis of (\s+[-\w|\p{InGreek}]*\s*){0,3} [CHEMICAL]
PRODUCT	(?i).* was used to prepare.* [CHEMICAL]
PRODUCT	(?i).* Giving \s [CHEMICAL]
PRODUCT	(?i).* Formation of \s [CHEMICAL]
PRODUCT	(?i).* One-pot synthesis of \s [CHEMICAL]
PRODUCT	(?i).* Preparation of (\s+[-\w|\p{InGreek}]*\s*){0,2} [CHEMICAL]
PRODUCT	(?i).* Yielding \s [CHEMICAL]
PRODUCT	(?i).* Leading to \s [CHEMICAL]
PRODUCT	(?i).* To afford (\s+[-\w|\p{InGreek}]*\s*) [CHEMICAL]
PRODUCT	[CHEMICAL] (?i).* were obtained from.*
PRODUCT	(?i).* To obtain (\s+[-\w|\p{InGreek}]*\s*){0,2} [CHEMICAL]
NONE, PRODUCT	[CHEMICAL] \s represent a new class of \s [CHEMICAL]
NONE, PRODUCT	[CHEMICAL] \s are.*building blocks for the synthesis of \s [CHEMICAL]
REACTAND	(?i).* dihydroxylation of (\s+[-\w|\p{InGreek}]*\s*){0,4} [CHEMICAL]
REACTAND	(?i).* To react with \s [CHEMICAL]
REACTAND	(?i).* Oxidation of \s [CHEMICAL]
REACTAND	(?i).* Oxidi(s|z)ed by \s [CHEMICAL]
REACTAND	(?i).* Reduction of \s [CHEMICAL]
REACTAND	(?i).* Reduced by \s [CHEMICAL]
REACTAND	[CHEMICAL] \s as (\s+[-\w|\p{InGreek}]*\s*) substrate.*

REACTAND, REACTAND	(?i).* Reaction between \s [CHEMICAL] \s and \s [CHEMICAL]
REACTAND, REACTAND	[CHEMICAL] (\s+[-\w|\p{InGreek}]*\s*) be oxidi(s|z)ed by \s [CHEMICAL]
REACTAND	[CHEMICAL] \s was oxidi(s|z)ed
REACTAND, REACTAND	[CHEMICAL] \s was treated with \s [CHEMICAL]
REACTAND, REACTAND	(?i).* Treatment of (\s+[-\w|\p{InGreek}]*\s*){0,2} [CHEMICAL]\s with(\s+[-\w|\p{InGreek}]*\s*){0,2} [CHEMICAL]
REACTAND, PRODUCT	(?i).* Transformation of \s [CHEMICAL] \s to \s [CHEMICAL] \s by
REACTAND, PRODUCT, REACTAND	(?i).* Transformation of \s [CHEMICAL] \s to \s [CHEMICAL] \s with \s [CHEMICAL]
PRODUCT, REACTAND, PRODUCT	[CHEMICAL] \s were obtained from \s [CHEMICAL] \s
REACTAND, PRODUCT	[CHEMICAL] \s was converted into \s [CHEMICAL]
CATALYST	(?i).* Catalytic amount of \s [CHEMICAL]
CATALYST	(?i).* In the presence of \s [CHEMICAL]
CATALYST	(?i).* Catalysis with \s [CHEMICAL]
CATALYST	[CHEMICAL] \s as a catalyst of.*
SOLVENT	(?i).* Extracted with \s [CHEMICAL]
SOLVENT	(?i).* In refluxing \s [CHEMICAL]
NONE, SOLVENT	[CHEMICAL] \s was dissolved in \s [CHEMICAL]

Appendix B

Curriculum Vitae

Born on 1980/07/15 in Hannover, Germany

Apr. 2008 – *current*	**L3S Research Center/Technical University of Braunschweig** PhD Student in computer science, research associate and teaching assistance
Apr. 2006 – Apr. 2008	**University of Hannover** Studies in computer science (Master of Science)
Oct. 2002 – Apr. 2006	**University of Hannover** Studies in computer science (Bachelor of Science)

List of Figures

List of Tables

Bibliography

[1] L. Candela, D. Castelli, N. Ferro, G. Koutrika, C. Meghini, P. Pagano, S. Ross, D. Soergel, M. Agosti, and M. Dobreva, Eds., *The DELOS Digital Library Reference model. Foundations for digital Libraries (Version 0.98)*. Pisa: ISTI-CNR at Gruppo ALI, 2008.

[2] J. R. McDaniel and J. R. Balmuth, "Kekule: OCR-optical chemical (structure) recognition," *Journal of Chemical Information and Modeling*, vol. 32, no. 4, pp. 373–378, Jul. 1992.

[3] A. T. Valko and A. P. Johnson, "CLiDE Pro: the latest generation of CLiDE, a tool for optical chemical structure recognition.," *Journal of Chemical Information and Modeling*, vol. 49, no. 4, pp. 780–7, Apr. 2009.

[4] M. Zimmermann, L. T. Bui Thi, and M. Hofmann, "Combating Illiteracy in Chemistry: Towards Computer-Based Chemical Structure Reconstruction," *ERCIM News*, no. 60, pp. 40–41, 2005.

[5] I. V Filippov and M. C. Nicklaus, "Optical Structure Recognition Software To Recover Chemical Information: OSRA, An Open Source Solution.," *Journal of Chemical Information and Modeling*, vol. 49, no. 3, pp. 740–3, Mar. 2009.

[6] P. Corbett and P. Murray-Rust, "High-throughput identification of chemistry in life science texts," in *Proceedings of the 2nd International Symposium on Computational Life Sciences*, 2006, vol. 4216, pp. 107–118.

[7] D. M. Jessop, S. E. Adams, E. L. Willighagen, L. Hawizy, and P. Murray-Rust, "OSCAR4: a flexible architecture for chemical text-mining.," *Journal of Cheminformatics*, vol. 3, no. 1, p. 41, Jan. 2011.

[8] T. Rocktäschel, M. Weidlich, and U. Leser, "ChemSpot: a hybrid system for chemical named entity recognition.," *Bioinformatics (Oxford, England)*, vol. 28, no. 12, pp. 1633–40, Jun. 2012.

[9] C. Kolárik, R. Klinger, C. M. Friedrich, M. Hofmann-Apitius, and J. Fluck, "Chemical names: terminological resources and corpora annotation," in *Workshop on Building and Evaluating Resources for Biomedical Text Mining (6th Edition of the Language Resources and Evaluation Conference)*, 2008, pp. 51–58.

[10] R. Hoffmann and P. Laszlo, "Representation in Chemistry," *Angewandte Chemie International Edition in English*, vol. 30, no. 1, pp. 1–16, 1991.

[11] J. Barnard, "Substructure searching methods: old and new," *Journal of Chemical Information and Computer Science*, pp. 532–538, 1993.

[12] G. M. Downs and P. Willett, "Similarity Searching in Databases of Chemical Structures.," in in *Reviews in Computational Chemistry, Volume 7*, 2009.

[13] P. Willett, J. M. Barnard, and G. M. Downs, "Chemical Similarity Searching," *Journal of Chemical Information and Modeling*, vol. 38, no. 6, pp. 983–996, Nov. 1998.

[14] C. G. Wermuth, C. R. Ganellin, P. Lindberg, and L. A. Mitscher, "Glossary of terms used in medicinal chemistry," *Pure and Applied Chemistry*, vol. 70, no. 5, pp. 1129–1143, 1998.

[15] Y. C. Martin and P. Willett, *Designing bioactive molecules: three-dimensional techniques and applications*. 1998, p. 276.

[16] "Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities." 2003.

[17] P. Mitra, C. Giles, B. Sun, and Y. Liu, "Chemxseer: a digital library and data repository for chemical kinetics," in *1st ACM Workshop on Cyber Infrastructure: Information Management in eScience*, 2007, pp. 7–10.

[18] J. H. Chen, E. Linstead, S. J. Swamidass, D. Wang, and P. Baldi, "ChemDB update--full-text search and virtual chemical space.," *Bioinformatics (Oxford, England)*, vol. 23, no. 17, pp. 2348–51, Oct. 2007.

[19] B. Sun, Q. Tan, P. Mitra, and C. L. Giles, "Extraction and search of chemical formulae in text documents on the web," in *Proceeding of the 16th International Conference on World Wide Web (WWW)*, 2007, pp. 251–260.

[20] B. Sun, P. Mitra, and C. L. Giles, "Mining, indexing, and searching for textual chemical molecule information on the web," in *Proceeding of the 17th International Conference on World Wide Web (WWW)*, 2008, pp. 735–744.

[21] S. Teufel, J. Carletta, and M. Moens, "An annotation scheme for discourse-level argumentation in research articles," in *Proceedings of the 9th Conference on European Chapter of the Association for Computational Linguistics*, 1999, p. 110.

[22] M. Liakata and L. Soldatova, "Semantic Annotation of Papers: Interface & Enrichment Tool (SAPIENT)," in *Proceedings of the BioNLP 2009 Workshop*, 2009, no. June, pp. 193–200.

[23] H. L. Morgan, "The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service.," *Journal of Chemical Documentation*, vol. 5, no. 2, pp. 107–113, 1965.

[24] D. J. Gluck, "A Chemical Structure Storage and Search System Developed at Du Pont.," *Journal of Chemical Documentation*, vol. 5, no. 1, pp. 43–51, Feb. 1965.

[25] E. G. Smith and P. A. Baker, Eds., *The Wiswesser Line-Formula Chemical Notation (WLN)*, 3rd ed. Cherry Hill, N. J.: Chemical Information Management, 1976.

[26] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *Journal of Chemical Information and Modeling*, vol. 28, no. 1, pp. 31–36, 1988.

[27] J. Barnard, C. Jochum, and S. Welford, "ROSDAL: A universal structure/substructure representation for PC-host communication," in in *Chemical Structure Information Systems: Interfaces, Communication and Standards, ACS Symposium Series No. 400*, W. Warr, Ed. Washington, DC: American Chemical Society, 1989, pp. 76–81.

[28] S. Ash, M. a. Cline, R. W. Homer, T. Hurst, and G. B. Smith, "SYBYL Line Notation (SLN): A Versatile Language for Chemical Structure Representation," *Journal of Chemical Information and Modeling*, vol. 37, no. 1, pp. 71–79, Jan. 1997.

[29] S. E. Stein, S. R. Heller, and D. Tchekhovskoi, "An Open Standard For Chemical Structure Representation: The IUPAC Chemical Identifier," in *Proceedings Of The International Chemical Information Conference*, 2003, pp. 131–143.

[30] J. A. Townsend, S. E. Adams, C. A. Waudby, V. K. de Souza, J. M. Goodman, and P. Murray-Rust, "Chemical documents: machine understanding and automated information extraction," *Journal of Organic & Biomolecular Chemistry*, vol. 2, no. 22, pp. 3294–3300, 2004.

[31] B. Sun, P. Mitra, C. Lee Giles, and K. T. Mueller, "Identifying, Indexing, and Ranking Chemical Formulae and Chemical Names in Digital Documents," *ACM Transactions on Information Systems*, vol. 29, no. 2, pp. 1–38, Apr. 2011.

[32] J. Klekota, F. P. Roth, and S. L. Schreiber, "Query Chem: a Google-powered web search combining text and chemical structures.," *Bioinformatics (Oxford, England)*, vol. 22, no. 13, pp. 1670–3, Jul. 2006.

[33] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, and E. Willighagen, "The Chemistry Development Kit (CDK): an open-source Java library for Chemo- and Bioinformatics.," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 2, pp. 493–500, 2003.

[34] C. Steinbeck, C. Hoppe, S. Kuhn, M. Floris, R. Guha, and E. L. Willighagen, "Recent Developments of the Chemistry Development Kit (CDK) - An Open-Source Java Library for Chemo- and Bioinformatics," *Journal of Current Pharmaceutical Design*, vol. 12, no. 17, pp. 2111–2120, Jun. 2006.

[35] L. H. Hall and L. B. Kier, "Electrotopological state indices for atom types: A novel combination of electronic, topological, and valence state information," *Journal of Chemical Information and Computer Sciences*, vol. 35, no. 6, pp. 1039–1045, 1995.

[36] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse, "Reoptimization of MDL Keys for Use in Drug Discovery," *Journal of Chemical Information and Modeling*, vol. 42, no. 6, pp. 1273–1280, Nov. 2002.

[37] Z. Hubálek, "Coefficients of association and similarity, based on binary (presence-absence) data: An Evaluation," *Journal of Biological Reviews*, vol. 57, pp. 669–689, 1982.

[38] J. Holliday, C. Hu, and P. Willett, "Grouping of coefficients for the calculation of inter-molecular similarity and dissimilarity using 2D fragment bit-strings," *Journal of Combinatorial Chemistry; High Throughput Screening*, vol. 5, no. 2, pp. 155–166, 2002.

[39] R. M. Cormack, "A Review of Classification," *Journal of the Royal Statistical Society. Series A (General)*, vol. 134, no. 3, pp. 321–367, 1971.

[40] L. A. Goodman and W. H. Kruskal, "Measures of Association for Cross Classifications," *Journal of the American Statistical Association*, vol. 49, no. 268, pp. 732–764, 1954.

[41] L. A. Goodman and W. H. Kruskal, "Measures of Association for Cross Classifications. II: Further Discussion and References," *Journal of the American Statistical Association*, vol. 54, no. 285, pp. 123–163, 1959.

[42] L. A. Goodman and W. H. Kruskal, "Measures of Association for Cross Classifications III: Approximate Sampling Theory," *Journal of the American Statistical Association*, vol. 58, no. 302, pp. 310–364, 1963.

[43] P. Willett, "Similarity-based approaches to virtual screening," *Journal of Biochemical Society Transactions*, vol. 31, pp. 603–606, Jun. 2003.

[44] M. G. Kendall, "A New Measure of Rank Correlation," *Journal of Biometrika*, vol. 30, no. 1–2, pp. 81–93, 1938.

[45] M. Hall, E. Frank, and G. Holmes, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[46] K. Degtyarenko, P. de Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcántara, M. Darsow, M. Guedj, and M. Ashburner, "ChEBI: a database and ontology for chemical entities of biological interest.," *Nucleic acids research*, vol. 36, no. Database issue, pp. D344–50, Jan. 2008.

[47] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais, "The vocabulary problem in human-system communication," *Communications of the ACM*, vol. 30, no. 11, pp. 964–971, Nov. 1987.

[48] R. Kraft and J. Zien, "Mining anchor text for query refinement," in *Proceedings of the 13th International Conference on World Wide Web (WWW)*, 2004, pp. 666–674.

[49] T. Haveliwala, "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 784–796, 2003.

[50] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, Mar. 2002.

[51] N. Belkin, "Interaction with Texts: Information Retrieval as Information-Seeking Behavior.," *Information Retrieval*, 1993.

[52] P. Ingwersen and N. Belkin, "Information retrieval in context-IRiX: workshop at SIGIR 2004-Sheffield," *ACM SIGIR Forum*, 2004.

[53] R. C. T. Morris, "Toward a user-centered information service," *Journal of the American Society for Information Science*, vol. 45, no. 1, pp. 20–30, Jan. 1994.

[54] T. Park, "Toward a theory of user-based relevance: A call for a new paradigm of inquiry," *Journal of the American Society for Information Science*, no. 1973, 1994.

[55] L. Schamber, M. B. Eisenberg, and M. S. Nilan, "A re-examination of relevance: toward a dynamic, situational definition∗," *Information Processing & Management*, vol. 26, no. 6, pp. 755–776, Jan. 1990.

[56] Z. Qiu and A. Doherty, "Mining user activity as a context source for search and retrieval," in *International Conference on Semantic Technology and Information Retrieval*, 2011, no. June, pp. 162–166.

[57] T. Maekawa, Y. Yanagisawa, Y. Sakurai, Y. Kishino, K. Kamei, and T. Okadome, "Context-aware web search in ubiquitous sensor environments," *ACM Transactions on Internet Technology*, vol. 11, no. 3, pp. 1–23, Jan. 2012.

[58] M. Voorhees, "Query Expansion using Lexical-Semantic Relations," *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1994.

[59] J. Xu and W. Croft, "Query expansion using local and global document analysis," in *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1996, pp. 4–11.

[60] R. Korfhage, "Query enhancement by user profiles," in *Proceedings of the 7th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1984.

[61] J. J. Rocchio, "Relevance feedback in information retrieval," *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 313–323, 1971.

[62] H. Keskustalo, K. Järvelin, and A. Pirkola, "Evaluating the effectiveness of relevance feedback based on a user simulation model: effects of a user scenario on cumulated gain value," *Information Retrieval*, vol. 11, no. 3, pp. 209–228, Jan. 2008.

[63] P. Anick, "Using Terminological Feedback for Web Search Refinement - A Log-based Study," pp. 88–95, 2003.

[64] D. Jiang, K. W. Leung, and W. Ng, "Context-aware search personalization with concept preference," in *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, 2011, no. 1, p. 563.

[65] J. Bai, J.-Y. Nie, G. Cao, and H. Bouchard, "Using query contexts in information retrieval," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2007, p. 15.

[66] J. Bai, D. Song, P. Bruza, J.-Y. Nie, and G. Cao, "Query expansion using term relationships in language models for information retrieval," *Proceedings of International Conference on Information and Knowledge Management (CIKM)*, p. 688, 2005.

[67] E. Meij, D. Trieschnigg, M. de Rijke, and W. Kraaij, "Conceptual language models for domain-specific retrieval," *Information Processing & Management*, vol. 46, no. 4, pp. 448–469, Jul. 2010.

[68] R. Kraft, C. C. Chang, F. Maghoul, and R. Kumar, "Searching with context," in *Proceedings of the 15th International Conference on World Wide Web (WWW)*, 2006, p. 477.

[69] L. Chen and Y. Papakonstantinou, "Context-sensitive ranking for document retrieval," in *Proceedings of ACM SIGMOD Conference*, 2011.

[70] S. H. Nguyen and W. Swieboda, "Extended Document Representation for Search Result Clustering," *Studies in Computational Intelligence*, pp. 77–95, 2012.

[71] R. Laza, R. Pavón, M. Reboiro-Jato, and F. Fdez-Riverola, "Evaluating the effect of unbalanced data in biomedical document classification.," *Journal of Integrative Bioinformatics*, vol. 8, no. 3, p. 177, Jan. 2011.

[72] F. Camous, S. Blott, and A. F. Smeaton, "Ontology-based MEDLINE document classification," in *Proceedings of the 1st International Conference on Bioinformatics Research and Development*, 2007.

[73] D. Trieschnigg, P. Pezik, V. Lee, F. de Jong, W. Kraaij, and D. Rebholz-Schuhmann, "MeSH Up: effective MeSH text classification for improved document retrieval.," *Bioinformatics (Oxford, England)*, vol. 25, no. 11, pp. 1412–8, Jun. 2009.

[74] S. Yoo and J. Choi, "Improving MEDLINE document retrieval using automatic query expansion," in *Proceedings of the 10th International Conference on Asian Digital Libraries (ICADL)*, 2007, pp. 241–249.

[75] S. Sie and J. Yeh, "Automatic Ontology Generation Using Schema Information," *Proceedings of the International Conference on Web Intelligence (WI)*, pp. 526–531, Dec. 2006.

[76] Y. Zhen and C. Li, "Cross-domain knowledge transfer using semi-supervised classification," in *Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, 2008, pp. 362–371.

[77] C. Liu, S. Wu, S. Jiang, and A. K. H. Tung, "Cross Domain Search by Exploiting Wikipedia," in *Proceedings of the International Conference on Data Engineering (ICDE)*, 2012, pp. 546–557.

[78] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 1606–1611.

[79] O. Egozi, S. Markovitch, and E. Gabrilovich, "Concept-Based Information Retrieval Using Explicit Semantic Analysis," *ACM Transactions on Information Systems (TOIS)*, vol. 29, no. 2, pp. 1–34, Apr. 2011.

[80] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou, "Exploiting Wikipedia as external knowledge for document clustering," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2009, p. 389.

[81] D. Milne and I. H. Witten, "An open-source toolkit for mining Wikipedia," *Journal of Artificial Intelligence*, vol. 194, pp. 222–239, Aug. 2012.

[82] D. Milne and I. Witten, "Learning to link with wikipedia," in *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, 2008.

[83] S. J. Nelson, D. Johnston, and B. L. Humphreys, "Relationships in Medical Subject Headings," in in *Relationships in the Organization of Knowledge*, C. A. Bean and R. Green, Eds. New York, NY, USA: Kluwer Academic Publishers, 2001, pp. 171–184.

[84] P. Wang, J. Hu, H.-J. Zeng, and Z. Chen, "Using Wikipedia knowledge to improve text classification," *Knowledge and Information Systems*, vol. 19, no. 3, pp. 265–281, Sep. 2008.

[85] J. Platt, "Fast Training of Support Vector Machines using Sequential Minimal Optimization," in in *Advances in Kernel Methods*, 1998.

[86] D. Milne, I. Witten, and D. Nichols, "A knowledge-based search engine powered by wikipedia," in *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM)*, 2007, p. 445.

[87] J. Surowiecki, "The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business," *Economies, Societies and Nations*, 2004.

[88] M. Sahlgren, "An Introduction to Random Indexing," in *Proceedings of the Methods and Applications of Semantic Indexing Workshop*, 2005, pp. 1–9.

[89] S. Tiun, "Automatic topic identification using ontology hierarchy," in *Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science*, 2001.

[90] J. Yu, J. A. Thom, and A. Tam, "Ontology evaluation using Wikipedia categories for browsing," in *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM)*, 2007, pp. 223–232.

[91] P. Schonhofen, "Identifying Document Topics Using the Wikipedia Category Network," *Proceedings of the International Conference on Web Intelligence (WI)*, pp. 456–462, Dec. 2006.

[92] E. Gabrilovich and S. Markovitch, "Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge," in *Proceedings of the National Conference on Artificial Intelligence*, 2006, vol. 21, no. 2, p. 1301.